

ОСНОВЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

**Практикум
к лабораторным занятиям по курсам
«Имитационное моделирование информационных процессов
и систем» и «Имитационное моделирование экономических
процессов» для студентов дневной формы обучения
специальностей 1-26 03 01 «Управление информационными
ресурсами» и 1-25 01 07 «Экономика и управление
на предприятии» специализации 1-25 01 07 02
«Экономическая информатика»**

УДК 303.094.7
ББК 65с51
О-75

Автор-составитель О. И. Еськова, канд. техн. наук, доцент

Рецензенты: И. В. Максимей, д-р техн. наук, профессор,
зав. кафедрой математических проблем управления
Гомельского государственного университета
им. Ф. Скорины;
Т. А. Заяц, ассистент кафедры информационно-
вычислительных систем Белорусского торгово-
экономического университета потребительской
кооперации

Рекомендован к изданию научно-методическим советом учреждения образования «Белорусский торгово-экономический университет потребительской кооперации». Протокол № 2 от 9 декабря 2008 г.

Основы имитационного моделирования : практикум к лабораторным занятиям по курсам «Имитационное моделирование информационных процессов и систем» и «Имитационное моделирование экономических процессов» для студентов дневной формы обучения специальностей 1-26 03 01 «Управление информационными ресурсами» и 1-25 01 07 «Экономика и управление на предприятии» специализации 1-25 01 07 02 «Экономическая информатика» / авт.-сост. О. И. Еськова. – Гомель : учреждение образования «Белорусский торгово-экономический университет потребительской кооперации», 2009. – 176 с.
ISBN 978-985-461-683-4

УДК 303.094.7
ББК 65с51

ISBN 978-985-461-683-4

© Учреждение образования «Белорусский
торгово-экономический университет
потребительской кооперации», 2009

ВВЕДЕНИЕ

Имитационное моделирование является в настоящее время наиболее активно развивающейся отраслью в области компьютерного моделирования. Оно широко используется для изучения процессов функционирования сложных систем, в том числе социально-экономических объектов и процессов. Наиболее хорошо разработаны средства имитационного моделирования систем массового обслуживания (СМО). Модели СМО применяются для решения задач управления в следующих сферах:

- *промышленности* при планировании сборочных операций, гибких автоматизированных производств; организации ремонта оборудования, работы инструментальных складов; расчете зон обслуживания в прядильном и ткацком производствах; определении числа резервных забоев в шахтах;
- *торговли* для определения количества магазинов, продавцов, фасовщиц, узлов расчета, торговых и кассовых автоматов;
- *службы быта* для расчета количества парикмахерских, пунктов приема коммунальных платежей, ремонтников в жилищно-эксплуатационных конторах, мастерских разного назначения, техники и работников в них;
- *транспорта* при анализе процессов дорожного движения, прохождения туннелей, очередей у светофоров, формирования железнодорожных составов, диспетчерских служб аэропортов, разгрузки и погрузки судов в портах, резервирования билетов и т. п.

Наиболее популярной системой имитационного моделирования СМО является система GPSS (General Purpose Simulating System) – общецелевая система моделирования, созданная Джеффри Гордоном в фирме «IBM» в начале 1960-х гг. Она является редким примером долгожительства в мире программного обеспечения. Фирмой «Minuteman Software» язык GPSS был перенесен на новую компьютерную платформу. Так появилась система GPSS World, которая ориентирована на современные персональные компьютеры и имеет стандартный оконный интерфейс. В настоящем пособии описывается студенческая версия GPSS World, которая отличается от коммерческой только количественным ограничением – до 150 блоков в составе модели.

Процесс исследования какого-либо экономического объекта с помощью системы GPSS заключается в разработке (программировании) модели, проведении имитационных экспериментов на ней и статистической оценке результатов этих экспериментов. При этом от исследователя требуется не только владение техникой программирования на языке GPSS, но и знание основных методов, связанных с планированием экспериментов, дисперсионным и регрессионным анализом, генерацией случайных чисел и другими аспектами имитационного моделирования. Сведения об этих методах рассматриваются в данном практикуме на конкретных примерах и по мере необходимости объясняются. В ходе выполнения лабораторных работ студенты получают представление о методе имитационного моделирования, областях его применения, преимуществах и ограничениях, а также овладевают начальными практическими навыками моделирования в системе GPSS World.

Первая лабораторная работа посвящена методу статистического моделирования Монте-Карло. Она несколько отличается от остальных, поскольку в качестве инструмента моделирования используются надстройки табличного процессора Microsoft Excel. Метод Монте-Карло рассматривается на примере задачи анализа инвестиционного проекта. При этом наглядно демонстрируются неправомерность использования средних показателей в задачах с вероятностным характером параметров исследуемой системы и возможные пути решения такого рода задач.

Вторая лабораторная работа имеет своей целью освоение студентами основных понятий языка моделирования GPSS на примере одноканальной системы массового обслуживания. Знакомство с принципами работы системы GPSS происходит через «ручное моделирование» на заданном примере. Ручной расчет показателей, которые в системе GPSS рассчитываются автоматически, позволяет лучше понять смысл этих показателей, методику их расчета и алгоритм работы системы имитации.

В третьей лабораторной работе рассматривается технология работы с системой моделирования GPSS World, а именно: интерфейс системы, диалоговые окна, структура стандартного отчета, простейшие приемы отладки моделей.

Четвертая и шестая лабораторные работы посвящены собственно приемам программирования моделей в GPSS. Рассматривается ряд простых примеров, которые были почерпнуты в основном из книги Т. Дж. Шрайбера «Моделирование на GPSS». Упор делается не столько на синтаксис тех или иных операторов GPSS, сколько на демонстрацию исследовательских возможностей системы. Каждая простая задача представляет собой, по сути дела, исследование, дающее ответ на вопрос, решение которого не является очевидным без моделирования. В то же время демонстрируются и приемы программирования, причем даются упражнения на создание или модификацию уже существующей модели с использованием каждого приема. В данных лабораторных работах студентам предлагается также ряд задач для самостоятельного выполнения, которые могут являться и вариантами индивидуальных заданий.

В пятой лабораторной работе обсуждаются вопросы генерации в GPSS случайных чисел с заданным законом распределения. Индивидуальные задания этой работы предлагают студентам сравнить результаты имитационного и аналитического моделирования простейших СМО различной структуры. Аналитические расчеты выполняются в пакете MathCad согласно методике, описанной в практикуме [8].

Седьмая лабораторная работа раскрывает особенности организации имитационного эксперимента, дает основные формулы расчета статистических характеристик исследуемых показателей, методику подготовки исходных данных для моделирования, методы автоматизации эксперимента и другие вопросы, связанные с эксплуатацией имитационных моделей. В качестве индивидуальных заданий студентам предлагается не просто выполнить программирование модели, но также грамотно осуществить автоматизированный эксперимент на этой модели с целью получения обоснованных и надежных оценок показателей.

Данный практикум может быть использован на занятиях по курсам «Имитационное моделирование экономических процессов», «Имитационное моделирование информационных процессов и систем», а также для самостоятельной подготовки студентов и аспирантов.

ЛАБОРАТОРНАЯ РАБОТА 1 ОЦЕНКА ИНВЕСТИЦИОННЫХ ПРОЕКТОВ МЕТОДОМ МОНТЕ-КАРЛО

1.1. Методы оценки инвестиционных проектов

Под инвестиционным проектом понимается любое вложение денежных средств, генерирующее денежные потоки в будущем. Примерами инвестиционных проектов могут служить закупка производственного оборудования, вложение денег в банк под процент, приобретение ценных бумаг. Для оценки инвестиционного проекта должны быть рассчитаны свободные денежные потоки по годам C_k (чистая прибыль в k -м году за вычетом расходов и налогов). Рассмотрим проект, начальные инвестиции в который составляют I_0 денеж. ед. (инвестиции в нулевом периоде). В конце k -го ($k = \overline{1, n}$) временного периода (например, года) проект приносит прибыль C_k денеж. ед. (табл. 1).

Таблица 1. Денежные потоки проекта

Номер временного периода	0	1	2	...	k	...	n
Денежный поток	$-I_0$	C_1	C_2	...	C_k	...	C_n

Одним из основных показателей эффективности инвестиционного проекта является *чистая текущая стоимость проекта (NPV)* [8]. Эта величина показывает, насколько меньше начальных инвестиций нужно вложить в данный проект по сравнению с альтернативным, при условии, что альтернативный проект даст денежные потоки, аналогичные потокам исследуемого проекта. Величина NPV рассчитывается по формуле

$$NPV = \sum_{k=1}^n \frac{C_k}{(1+r)^k} - I_0, \quad (1)$$

где r – норма дисконтирования денежных потоков проекта (внутренняя норма прибыли альтернативных проектов).

Если величина $NPV > 0$, то проект считается выгодным для инвестирования, а если $NPV < 0$, то выгоднее вкладывать деньги в альтернативные проекты. При сравнении нескольких инвестиционных проектов с одинаковым уровнем риска предпочтительным является тот, который имеет большее значение NPV .

Денежный поток проекта зависит от многих факторов, например, от цены на продукцию, от объемов ее выпуска, цен на ресурсы и энергоносители, налоговой ставки и пр. В любом случае все эти факторы являются случайными величинами и со временем могут изменяться. Поэтому денежный поток в k -м периоде можно рассматривать как случайную величину, имеющую математическое ожидание m_k и среднеквадратическое отклонение σ_k . Тогда и значение чистой текущей стоимости (NPV) также будет являться случайной величиной со своим математическим ожиданием m и среднеквадратическим отклонением σ . Причем значение σ характеризует разброс возможных значений NPV и поэтому может служить одной из оценок риска инвестиционного проекта.

Оценка статистических характеристик величины NPV с помощью аналитических методов является весьма трудной и часто нереализуемой задачей. Поэтому часто для этой цели используется метод статистических испытаний (*метод Монте-Карло*). Название этого метода связано с известным центром игорного бизнеса Монте-Карло, поскольку рулетка может являться простейшим примером генератора случайных чисел. Суть метода Монте-Карло состоит в проведении ряда статистических экспериментов, в каждом из которых разыгрывается определенное значение случайных факторов, влияющих на результат. В совокупности полученные значения этих факторов дают случайную реализацию величины результирующего показателя. При достаточно большом количестве статистических экспериментов и последующей обработке их результатов, можно получить такие статистические характеристики результирующего показателя, как математическое ожидание, среднеквадратическое отклонение и др. Можно также анализировать различные гипотезы, касающиеся закона распределения этой случайной величины.

В данной лабораторной работе предлагается оценить чистую текущую стоимость проекта в ситуации, когда имеются два случайных фактора, оказывающих влияние на его величину: цена и переменные затраты на единицу готовой продукции. Сначала чистая текущая стоимость будет рассчитана на основании наиболее вероятных значений этих величин, а затем будут получены ее оценки с помощью метода Монте-Карло. Студентам предлагается сравнить результаты этих расчетов.

1.2. Описание моделируемого объекта

Рассмотрим инвестиционный проект, рассчитанный на 5 лет ($n = 5$). Начальные инвестиции в этот проект составляют $I_0 = 2\,000$ денеж. ед. Проект предусматривает производство нового вида продукции ежегодно в количестве $Q = 200$ ед. Цена этой продукции (P) и переменные затраты на единицу продукции (V), которые включают стоимость сырья, являются случайными величинами. Эксперты оценили эти величины так, как показано в табл. 2.

Таблица 2. Экспертные оценки случайных величин

Характеристики	Переменные затраты (V)	Цена за единицу (P)	Вероятность
Минимум	25	40	0,25
Наиболее вероятное	30	50	0,50
Максимум	35	55	0,25

Кроме этих случайных факторов, на денежные потоки проекта влияют и другие величины, которые мы для простоты будем считать детерминированными (неизменяющимися): это постоянные годовые расходы ($F = 500$), амортизация ($A = 100$), налоговая ставка ($T = 60\%$). Норма дисконтирования ($r = 0,1$) также считается неизменной.

Для расчета свободного денежного потока за каждый год нужно определить следующие показатели:

- прибыль от продаж: $ПП = Q(P - V)$;
- налогооблагаемую прибыль: $НОП = ПП - F - A$;
- величину уплачиваемого налога: $H = НОП \cdot T$;
- чистую прибыль: $ЧП = НОП - H$;
- свободный денежный поток: $C_k = ЧП + A$.

Таким образом, итоговая формула для расчета свободного денежного потока будет следующая:

$$C_k = (Q(P - V) - F - A)(1 - T) + A. \quad (2)$$

Будем считать, что денежные потоки в каждом периоде (от 1 до 5) одинаковы, поскольку одинаков объем производства продукции. Мы не рассматриваем возможность изменения параметров системы с течением времени, но если моделировать ситуацию, приближенную к действительности, то возможность изменения факторов во времени должна учитываться.

1.3. Расчет чистой текущей стоимости проекта на основании наиболее вероятных значений

На первом листе книги Excel введите исходные данные по проекту, как показано на рис. 1.

	A	B	C	D
1	Аналитический расчет (по наиболее вероятному)			
2	Начальные инвестиции (I ₀)	2000	Норма дисконтирования (r)	0,1
3	Постоянные расходы (F)	500	Налог (T)	0,6
4	Амортизация (A)	100	Срок (n)	5
5	Количество продукции (Q)	200		
6		переменные расходы (V)	цена (P)	вероятность
7	минимум	25	40	0,25
8	вероятное	30	50	0,5
9	максимум	35	55	0,25

Рис 1. Исходные данные по проекту

Далее осуществите расчет свободного денежного потока, используя в качестве V и P их наиболее вероятные значения. Расчеты расположите в таблице, как показано на рис. 2. Используйте формулы предыдущего раздела либо сразу запишите формулу (2).

	А	В	С	Д	Е
11	Расчет ежегодного денежного потока				
12	Прибыль от продаж	Налогооблагаемая прибыль	Налог	Чистая прибыль	Свободный денежный поток
13	4000	3400	2040	1360	1460

Рис. 2. Расчет ежегодного денежного потока

Далее постройте таблицу денежных потоков проекта (рис. 3). Для свободного денежного потока в нулевом периоде запишите ссылку на ячейку В2 (значение I_0) со знаком минус, а для остальных денежных потоков – ссылку на ячейку Е13, которая содержит значение рассчитанного свободного денежного потока.

Дисконтированный денежный поток для k -го года рассчитывается по формуле

$$PV_k = \frac{C_k}{(1+r)^k}. \quad (3)$$

Формулу (3) внесите в ячейки строки 19 (кроме нулевого периода, для которого дисконтированный денежный поток равен свободному).

Чистая текущая стоимость проекта в ячейке Н19 рассчитывается как сумма дисконтированных денежных потоков. Легко заметить, что описанная последовательность расчетов реализует формулу (1).

	А	В	С	Д	Е	Ф	Г	Н
17	Номер года	0	1	2	3	4	5	
18	Свободный денежный поток	-2000	1460	1460	1460	1460	1460	Чистая текущая стоимость проекта
19	Дисконтированный денежный поток	-2000	1327,27	1206,61	1096,92	997,20	906,55	3534,5487

Рис. 3. Таблица денежных потоков проекта

Однако, для проекта, который дает одинаковые денежные потоки по всем периодам, существует более простой способ расчета NPV , не требующий построения таблицы денежных потоков. Можно использовать стандартную функцию *Приведенная стоимость* ПС() (категория *Финансовые*)*, которая возвращает текущую стоимость инвестиции, равноценную ряду будущих выплат. Введите эту функцию в ячейку F13. Она имеет следующие параметры:

$$ПС(Ставка, КПер, Плт, Бс, Тип),$$

где *Ставка* – процентная ставка за период (используйте значение нормы дисконтирования r);

КПер – общее число периодов выплат инвестиций (используйте срок проекта n);

Плт – выплата в каждый период (используйте значение рассчитанного свободного денежного потока со знаком минус);

Бс – будущая стоимость после последней выплаты (имеет значение 0);

Тип – логическое значение, которое равно нулю, если выплаты производятся в конце периода, и единице, если вначале (задайте значение 0).

Чтобы получить значение чистой текущей стоимости, нужно из результата расчета по этой функции вычесть величину начальных инвестиций (значение I_0). Таким образом, формула в ячейке F13 будет иметь следующий вид:

$$=ПС(D2;D4;-E13;0;0)-B2$$

Рассчитанное по этой формуле значение NPV должно совпадать со значением в ячейке Н19.

1.4. Применение метода Монте-Карло для оценки характеристик чистой текущей стоимости проекта

Скопируйте на второй лист Excel исходные данные по проекту, которые показаны на рис. 1. Заголовок в первой строке исправьте на «Имитационный анализ (метод Монте-Карло)».

* В пакете Microsoft Excel 98 эта функция называется ПЗ().

Для генерации случайных значений величин V и P необходимо рассчитать их математическое ожидание и среднеквадратическое отклонение. Математическое ожидание случайной величины X рассчитывается по формуле

$$M(X) = \sum_{i=1}^n p_i \cdot X_i, \quad (4)$$

где X_i – значение случайной величины, которое наблюдается с вероятностью p_i .

Исходя из этой формулы, в строке 11 для расчета математического ожидания внесите функцию СУММПРОИЗВ() (рис. 4).

	А	В	С	Д
6		переменные расходы (V)	цена (P)	вероятность
7	минимум	25	40	0,25
8	вероятное	30	50	0,5
9	максимум	35	55	0,25
10				
11	среднее	=СУММПРОИЗВ(B7:B9;\$D\$7:\$D\$9)		=СУММПРОИЗВ(B7:B9;D7:D9)
12	станд. отклон.	=КОРЕНЬ(СУММПРОИЗВ((B7:B9-B11)^2;\$D\$7:\$D\$9))		=КОРЕНЬ(СУММПРОИЗВ((B7:B9-B11)^2;D7:D9))

Рис. 4. Расчет характеристик входных случайных величин

Среднеквадратическое отклонение случайной величины X рассчитывается по следующей формуле:

$$\sigma(X) = \sqrt{M(X - M(X))^2} = \sqrt{\sum_{i=1}^n p_i (X_i - M(X))^2}. \quad (5)$$

Внесите соответствующую формулу Excel в строку 12:

$$=КОРЕНЬ(СУММПРОИЗВ((B7:B9-B11)^2;D7:D9)).$$

Результаты расчета по этим формулам показаны в строках 11 и 12 на рис. 5.

Далее сформируйте шапку таблицы, в которую будут занесены результаты статистических экспериментов (рис. 5).

	А	В	С	Д	Е	Ф	Г	Н	И
11	среднее	30	48,75						
12	станд. отклон.	3,54	5,45						
13									
14	Эксперименты								
15	№	Переменные расходы	Цена	Прибыль от продаж	Налогооблагаемая прибыль	Налог	Чистая прибыль	Свободный денежный поток (Ск)	Чистая текущая стоимость проекта (NPV)

Рис. 5. Шапка таблицы для проведения статистических экспериментов

Планируется проведение 20-ти экспериментов, поэтому в первый столбец занесите номера от 1 до 20. Будем считать, как уже указывалось ранее, что значения исследуемых факторов не изменяются с течением времени. Поэтому для каждого расчета NPV необходимо иметь одно случайное значение переменных расходов (в столбце В) и одно случайное значение цены производимой продукции (в столбце С). Предположим, что переменные расходы и цена имеют нормальный закон распределения с математическим ожиданием и среднеквадратическим отклонением, которые были рассчитаны в строках 11 и 12. Для розыгрыша случайных величин с такими характеристиками воспользуемся надстройкой *Пакет анализа*.

Задайте команду **Сервис/Анализ данных** и выберите инструмент анализа *Генерация случайных чисел*. Заполните появившееся окно, как показано на рис. 6.

Приведем необходимые пояснения. Поле *Число переменных* задает количество колонок таблицы, в которых будут размещаться сгенерированные случайные величины. Поскольку в нашем случае все случайные значения переменных затрат должны быть в одном столбце (В), то значение этого поля равно 1.

Поле *Число случайных чисел* задает количество чисел, которые должны быть сгенерированы (в нашем случае 20).

Необходимый вид распределения задается путем выбора его из раскрывающегося списка *Распределение*. Выбранный вид распределения определяет внешний вид диалогового окна. Для нормального распределения нужно задать его параметры: среднее значение (математическое ожидание) и стандартное (среднеквадратическое) отклонение. К сожалению, они могут быть заданы только числовыми константами, нельзя использовать ссылку на ячейку таблицы.

Указание аргумента *Случайное рассеивание* позволяет при повторных запусках генератора получать те же значения случайных величин, что и при первом. В случае, если этот аргумент не был задан, равен нулю, то при каждом последующем запуске генератора будет формироваться новая выборка из генеральной сово-

купности. В нашем случае значение 1 в этом поле позволит оперировать с одним и тем же набором величин и избежать постоянных пересчетов таблицы.

Последний аргумент этого окна – *Параметры вывода* задает место расположению результатов. Установите переключатель в положение *Выходной интервал* и укажите левый верхний угол диапазона, в котором будет размещен результат (щелкните мышью по ячейке B16).

Рис. 6. Заполнение полей окна *Генерация случайных чисел*

После нажатия кнопки *OK* будет заполнен столбец случайных значений переменных расходов в таблице экспериментов. Аналогичным способом получите столбец случайных значений для цены продукции.

Далее введите формулы расчета свободных денежных потоков и *NPV*, как это было описано в разделе 1.3. Скопируйте эти формулы методом автозаполнения для всех остальных строк экспериментов. Пример возможных результатов показан на рис. 7.

14	Эксперименты								
	№	Переменные расходы	Цена	Прибыль от продаж	Налогооблагаемая прибыль	Налог	Чистая прибыль	Свободный денежный поток (Ск)	Чистая текущая стоимость проекта (NPV)
15									
16	1	19,299	32,275	2595,208	1995,208	1197,125	798,083	898,083	1 404,44р.
17	2	30,567	49,622	3811,145	3211,145	1926,687	1284,458	1384,458	3 248,19р.
18	3	26,935	44,031	3419,270	2819,270	1691,562	1127,708	1227,708	2 653,98р.
19	4	33,091	53,509	4083,587	3483,587	2090,152	1393,435	1493,435	3 661,29р.
20	5	30,760	49,920	3832,025	3232,025	1939,215	1292,810	1392,810	3 279,85р.
21	6	29,821	48,475	3730,720	3130,720	1878,432	1252,288	1352,288	3 126,23р.
22	7	28,639	46,654	3603,108	3003,108	1801,865	1201,243	1301,243	2 932,74р.
23	8	34,456	55,611	4230,891	3630,891	2178,535	1452,356	1552,356	3 884,65р.
24	9	33,279	53,798	4103,825	3503,825	2102,295	1401,530	1501,530	3 691,98р.
25	10	32,350	52,368	4003,588	3403,588	2042,153	1361,435	1461,435	3 539,99р.
26	11	26,679	43,638	3391,663	2791,663	1674,998	1116,665	1216,665	2 612,12р.
27	12	33,808	54,612	4160,873	3560,873	2136,524	1424,349	1524,349	3 778,48р.
28	13	31,964	51,774	3961,953	3361,953	2017,172	1344,781	1444,781	3 476,86р.
29	14	30,120	48,935	3762,963	3162,963	1897,778	1265,185	1365,185	3 175,13р.
30	15	28,184	45,954	3554,055	2954,055	1772,433	1181,622	1281,622	2 858,36р.
31	16	22,316	36,921	2920,872	2320,872	1392,523	928,349	1028,349	1 898,25р.
32	17	25,284	41,490	3241,113	2641,113	1584,668	1056,445	1156,445	2 383,84р.
33	18	28,773	46,861	3617,605	3017,605	1810,563	1207,042	1307,042	2 954,72р.
34	19	26,290	43,038	3349,653	2749,653	1649,792	1099,861	1199,861	2 548,42р.
35	20	26,564	43,461	3379,269	2779,269	1667,562	1111,708	1211,708	2 593,33р.

Рис. 7. Результаты статистических испытаний

Результаты экспериментов должны быть обработаны с целью получения статистических характеристик выходных, а заодно и входных, параметров по полученной выборке из генеральной совокупности. Рассчитаем только простейшие характеристики: среднее значение, стандартное отклонение, коэффициент вариации, минимальное и максимальное значения по формулам на рис. 8. Результаты расчетов по этим формулам показаны на рис. 9.

	A	B	C	D	E
37	Анализ результатов				
38	Показатели	Переменные расходы (V)	Цена (P)	Свободный денежный поток (C _j)	Чистая текущая стоимость проекта (NPV)
39	сред. значение	=СРЗНАЧ(B16:B35)	=СРЗНАЧ(C16:C35)	=СРЗНАЧ(D16:D35)	=СРЗНАЧ(E16:E35)
40	станд. отклон.	=СТАНДОТКЛОНП(B16:B35)	=СТАНДОТКЛОНП(C16:C35)	=СТАНДОТКЛОНП(D16:D35)	=СТАНДОТКЛОНП(E16:E35)
41	коэф. вариации	=B40/B39	=C40/C39	=D40/D39	=E40/E39
42	Минимум	=МИН(B16:B35)	=МИН(C16:C35)	=МИН(D16:D35)	=МИН(E16:E35)
43	Максимум	=МАКС(B16:B35)	=МАКС(C16:C35)	=МАКС(D16:D35)	=МАКС(E16:E35)

Рис. 8. Формулы статистического анализа

	A	B	C	D	E
37	Анализ результатов				
38	Показатели	Переменные расходы (V)	Цена (P)	Свободный денежный поток (C _j)	Чистая текущая стоимость проекта (NPV)
39	сред. значение	28,96	47,15	1315,07	2985,14
40	станд. отклон.	3,82	5,88	164,86	624,95
41	коэф. вариации	0,13	0,12	0,13	0,21
42	Минимум	19,30	32,27	898,08	1404,44
43	Максимум	34,46	55,61	1552,36	3884,65

Рис. 9. Результаты статистического анализа

Сформулируем выводы по результатам этих расчетов.

- Среднее значение чистой текущей стоимости проекта по результатам статистических испытаний составляет $NPV_2 = 2\,985,14$ денеж. ед. Это немного меньше значения, рассчитанного по наиболее вероятным значениям переменных расходов и цены продукции (см. лист 1 Excel) $NPV_1 = 3\,534,55$ денеж. ед. Необходимо отметить, что ни то, ни другое значение NPV нельзя считать достаточно точным. Расчет по наиболее вероятным значениям (NPV_1) не учитывает стохастического характера входных параметров V и P . А значение NPV_2 рассчитано только по результатам 20-ти статистических экспериментов (для простоты лабораторной работы), в реальной же практике имитационного моделирования требуется проведение около 500–1 000 экспериментов. Расчет необходимого числа экспериментов требует дополнительного анализа.

- Коэффициент вариации (отношение среднеквадратического отклонения к среднему значению) для чистой текущей стоимости проекта характеризует риск этого проекта. Его величина составляет 0,21, т. е. значительно меньше единицы, что свидетельствует о том, что риск данного проекта невысок.

- В ходе рассмотренных 20-ти экспериментов не встретилось ни одного отрицательного значения NPV , что свидетельствует о его выгодности для инвестора. Следует отметить, однако, что при большем числе экспериментов могут появиться и отрицательные значения этого показателя, причем их доля от общего числа экспериментов будет характеризовать вероятность того, что проект может стать невыгодным из-за колебания цен на продукцию и сырье.

1.5. Задание для самостоятельной работы

Исследуйте значения чистой текущей стоимости проекта для других законов распределения случайных величин V и P . Рассмотрите закон Пуассона и равномерное распределение, для простоты рассмотрите один и тот же закон для обоих факторов. Учтите, что для закона Пуассона нужно указать только среднее значение случайной величины, а для равномерного распределения – минимальное и максимальное возможные значения. Результаты оформите в тетради для лабораторных работ, как показано в табл. 3. Сделайте выводы.

Таблица 3. Форма оформления результатов самостоятельной работы

Варианты расчета	NPV (среднее значение)	Коэффициент вариации
По наиболее вероятным величинам	3 534,55	–
Нормальное распределение	2 985,14	0,21
Распределение Пуассона		
Равномерное распределение		

Контрольные вопросы

1. Что такое инвестиционный проект? Как оценивается его эффективность?
2. Какой показатель эффективности инвестиционного проекта обозначается аббревиатурой *NPV*? По какой формуле он рассчитывается?
3. Почему аналитические методы оценки инвестиционных проектов часто дают неудовлетворительные результаты?
4. В чем состоит суть метода Монте-Карло?
5. Как рассчитывается величина свободного денежного потока (C), если заданы: объем производимой продукции (Q), цена единицы продукции (P), переменные затраты на единицу продукции (V), постоянные расходы (F), амортизация (A) и налоговая ставка (T)?
6. Какая функция Excel может быть использована для расчета *NPV*, если проект дает одинаковые денежные потоки по годам?
7. Как рассчитывается математическое ожидание и среднеквадратическое отклонение, если случайная величина задана таблицей своих возможных значений и соответствующих вероятностей?
8. Какие функции Excel используются для расчета среднего значения и стандартного отклонения по выборке из генеральной совокупности?
9. Какая надстройка Excel используется для генерации случайных величин с заданным законом распределения? Какие параметры распределения нужно указывать:
 - а) для нормального закона;
 - б) для равномерного закона;
 - в) для закона Пуассона?
10. Как рассчитывается коэффициент вариации для *NPV* и что он характеризует?

ЛАБОРАТОРНАЯ РАБОТА 2 ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКА МОДЕЛИРОВАНИЯ GPSS

2.1. Описание моделируемого объекта

Основные понятия языка GPSS будем рассматривать на примере моделирования простой системы массового обслуживания с одним обслуживающим устройством и одной очередью. Таким обслуживающим устройством может быть продавец в маленьком магазине, билетер в театральной кассе, кладовщик на складе или центральный процессор в вычислительной системе. В литературе обслуживающее устройство может называться прибором или каналом обслуживания. Для определенности рассмотрим парикмахерскую с одним мастером. Обслуживающим устройством является парикмахер. Клиенты приходят в парикмахерскую в случайные моменты времени и ждут своей очереди на обслуживание, если в этом есть необходимость. Их обслуживают по принципу «первым пришел – первым обслужен», после чего они уходят. Схематично структура этой системы показана на рис. 10.



Рис. 10. Структура моделируемой системы

Предположим, что интервал времени между прибывающими в парикмахерскую клиентами есть случайная величина, имеющая равномерный закон распределения. Функция плотности распределения непрерывной случайной величины, равномерно распределенной на отрезке $[a, b]$, показана на рис. 11.

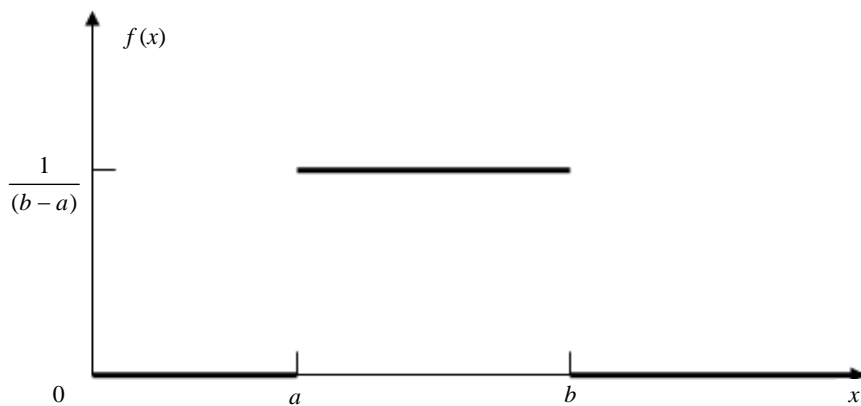


Рис. 11. Функция плотности распределения равномерной непрерывной случайной величины

Обычно равномерное распределение задается средним значением $\frac{a+b}{2}$ и размахом, т. е. половиной интервала варьирования $\frac{b-a}{2}$. Например, распределение от 12 до 24 запишется как 18 ± 6 .

В случае дискретного равномерного распределения все значения случайной величины из интервала $[a, b]$ имеют одинаковую вероятность.

Физическое значение единицы времени может быть выбрано разработчиком произвольно (1 сут, 1 мин, 1 с или 5 с), только необходимо следить, чтобы все данные, связанные со временем, были выражены через эту минимальную единицу времени.

Время обслуживания клиента парикмахером также есть равномерно распределенная случайная величина. Например, если время обслуживания может принимать значения от 12 до 20 с, то такое распределение будет представлено как (16 ± 4) с.

Целью моделирования является определение ряда следующих статистических величин:

- числа клиентов, обслуженных за данный промежуток времени, например, за рабочий день;
- числа клиентов, которые попали на обслуживание сразу же по прибытии, не ожидая в очереди;
- среднего времени пребывания клиента в очереди;
- средней длины очереди;
- максимальной длины очереди;
- коэффициента загрузки парикмахера, т. е. отношения времени, в течение которого парикмахер был занят обслуживанием, к общему времени моделирования.

2.2. События и таймер модельного времени

Для организации процесса моделирования во времени в системе GPSS используется специальная переменная *таймер модельного времени*. Эта переменная может изменяться только в сторону увеличения. В первых версиях языка GPSS предполагалось, что таймер модельного времени может принимать только целые значения. Данное ограничение отменено в системе GPSS World.

Следует различать такие понятия, как время моделирования и модельное время. Модельное время – это значение переменной *таймер модельного времени*, которое является образом реального времени моделируемой системы. А время моделирования – это время, затраченное компьютером на выполнение имитационного эксперимента, т. е. прогона модели. Очевидно, что время моделирования зависит не от того, какой реальный интервал времени имитируется, а от того, сколько событий происходит в системе.

Событием в системе массового обслуживания называется изменение состояния этой системы. В нашем случае их три: прибытие клиента, начало обслуживания и завершение обслуживания. Каждое событие происходит в определенный момент времени. Можно выделить две категории событий: основные и вспомогательные.

Основным называется такое событие, время возникновения которого в ходе моделирования можно запланировать заранее, т. е. рассчитать его до фактического возникновения. В нашем примере, основным событием является приход клиента в парикмахерскую. Моделирующий алгоритм GPSS после имитации поступления очередного клиента в парикмахерскую планирует приход следующего клиента, разыгрывая случайное число в соответствии с заданным распределением. Это значение можно интерпретировать как величину интервала времени, которое должно пройти прежде, чем появится следующий клиент. Тогда время появления следующего клиента может быть рассчитано как сумма текущего значения таймера модельного времени и разыгранного значения интервала (рис. 12).

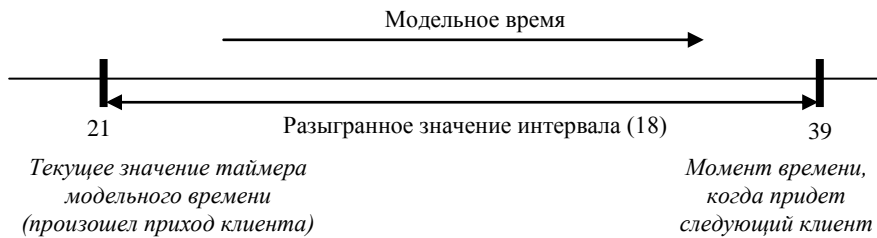


Рис. 12. Временная диаграмма планирования прихода очередного клиента

Основным также является событие завершения обслуживания клиента. Обслуживание клиента имитируется задержкой на время обслуживания. Завершение обслуживания планируется в тот момент, когда имитируется поступление нового клиента на обслуживание (он садится в кресло к парикмахеру), и выполняется аналогично процедуре планирования прихода следующего клиента (рис. 13).

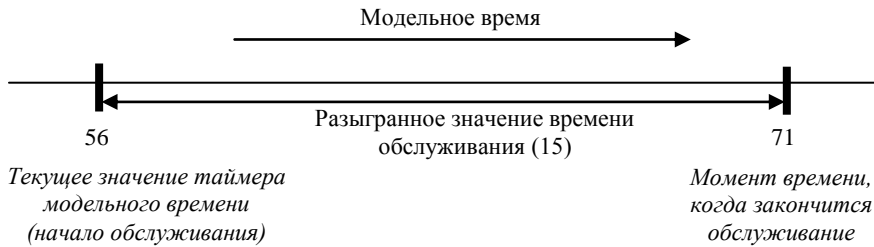


Рис. 13. Временная диаграмма планирования завершения обслуживания клиента

Вспомогательными называются события, время возникновения которых невозможно запланировать заранее. Они являются следствием основных и происходят в те же моменты времени, что и основные. Примером вспомогательного события является поступление клиента на обслуживание. Оно возникает, если ему непосредственно предшествует основное событие. Например, если возникло основное событие *Приход клиента* и парикмахер свободен, то возникает и вспомогательное событие *Поступление клиента на обслуживание*. Или если возникло основное событие *Окончание обслуживания* и имеются клиенты, ожидающие в очереди, тогда также происходит вспомогательное событие *Поступление клиента на обслуживание*.

Таким образом, события в системе происходят только в те моменты времени, когда запланированы основные события. Между ними состояние системы не изменяется. Поэтому для изменения таймера модельного времени в системе GPSS выбран метод переменного приращения значения таймера (принцип Δx). Суть его заключается в том, что значение таймера модельного времени после обработки события увеличивается до значения времени следующего ближайшего события.

Пусть, например, разыгранные значения интервалов времени между приходами клиентов составляют 14, 13, 17, 15, ...; а разыгранные значения времени обслуживания клиента составляют 18, 12, 14, Временная диаграмма изменения состояний системы показана на рис. 14. На этой диаграмме ось t_1 показывает время прихода клиентов в парикмахерскую, а ось t_2 показывает интервалы обслуживания клиентов парикмахером. Поскольку на ось t_2 пунктиром спроектированы моменты прихода клиентов, она фактически показывает все моменты времени, когда в системе происходят какие-либо события. Таким образом, из этой диаграммы очевидно, что таймер модельного времени будет последовательно получать следующие значения: 14, 27, 32, 44, 58, 59, В табл. 4 перечислены события, которые должны быть промоделированы в эти моменты времени.



Рис. 14. Временная диаграмма изменения состояний моделируемой системы

Таблица 4. Моменты наступления событий в моделируемой системе

Значения таймера модельного времени	События, происходящие в эти моменты времени
14	Приход первого клиента, поступление его на обслуживание
27	Приход второго клиента
32	Окончание обслуживания первого клиента, поступление второго клиента на обслуживание
44	Окончание обслуживания второго клиента, приход третьего клиента, поступление третьего клиента на обслуживание
58	Окончание обслуживания третьего клиента
59	Приход четвертого клиента, поступление его на обслуживание

2.3. Объекты языка GPSS

Выделяют четыре основные категории объектов GPSS (табл. 5).

Таблица 5. Объекты языка GPSS

Категория объекта	Тип объекта	Мнемоническое обозначение
Динамическая	Транзакт	–
Статическая, или оборудование	Устройство	F
	Память	S
	Ключ	L
Статистическая	Очередь	Q
	Таблица	T
Операционная	Блок	–

Динамическая категория объектов представлена одним типом – транзактом. Транзакт имитирует единицу исследуемого потока запросов на обслуживание. Например, он может представлять клиента, пришедшего в парикмахерскую или магазин; деталь, подлежащую обработке на станке; автомобиль в модели транспортной системы; задание пользователя в вычислительной системе и т. д. Транзакт во многом аналогичен запросу в системе массового обслуживания, но это понятие более широкое и сложное. Таким образом, *транзакт* – это динамический (движущийся) элемент GPSS-модели. Работа модели заключается в перемещении транзактов от одного операционного блока к другому. Именно транзакт, двигаясь по модели, является инициатором смены состояний моделируемой системы. Ему может быть присвоен приоритет, который определяет порядок обслуживания его в очереди. Транзакты с более высоким приоритетом выбираются на обслуживание в первую очередь. Также с каждым из транзактов может быть связан ряд параметров. Например, если транзакт имитирует собой судно, то параметром может быть количество груза, подлежащего разгрузке. Это значение используется для определения времени разгрузки у причала порта.

К статической категории (оборудованию) относятся устройства, памяти и ключи. *Устройства* (FACIL-ITY) используются для моделирования ресурса, который в каждый момент времени может быть занят только одним транзактом (одноканальная СМО), например, кассир, парикмахер или продавец, кладовщик на складе, центральный процессор в вычислительной системе и т. п. *Памяти* (STORAGE) представляют собой оборудование, которое может использоваться несколькими транзактами одновременно (многоканальная СМО), например, запоминающее устройство вычислительной системы, стоянка автомобилей с ограниченным числом мест, несколько причалов в порту и т. д. Логические *ключи* (SWITCH) служат для блокировки или изменения маршрута движения транзактов. Они могут быть использованы в модели, например, для изображения светофора или переключателя.

Статистические объекты – очереди и таблицы, предназначены для сбора статистических данных в различных точках модели и для представления их в виде стандартного отчета. *Очередь* (QUEUE) – это объект, предназначенный для сбора статистики по задержанным сообщениям (среднего времени ожидания, максимальной длины очереди и т. д.). Необходимо отметить, что объект *очередь* не влияет на создание очереди транзактов, а лишь служит для сбора статистики по тем очередям, которые нас интересуют. *Таблица* (TABLE) используется для сбора статистической информации в виде эмпирических функций распределения случайных величин, получаемых в ходе моделирования.

Все вышеперечисленные объекты присутствуют в модели неявно, т. е. они существуют в процессе моделирования в виде некоторых информационных структур (переменных, списков), к которым пользователь, или разработчик модели, имеет весьма ограниченный доступ. А непосредственно для разработки модели используются блоки.

Блоки относятся к категории операционных объектов и задают логику функционированию моделируемого процесса или системы, определяют пути движения транзактов между объектами категории оборудования. Модель представляет собой несколько отдельных сегментов, каждый из которых состоит из после-

довательности блоков и определяет путь движения некоторого типа транзактов. Все события в моделируемой системе происходят в результате входа транзактов в блоки и выполнения блоками своих функций. Основные функции блоков следующие:

- создание (генерация) и уничтожение транзактов;
- задержка транзакта на некоторый интервал времени;
- изменение маршрута движения транзактов;
- изменение числовых атрибутов объектов.

Каждому блоку соответствует оператор языка GPSS. Кроме того, программа модели может быть представлена в виде блок-схемы, на которой каждому блоку соответствует характерная фигура.

Транзакт, сгенерированный в каком-то блоке, перемещается между блоками в рамках своего сегмента. Каждый блок можно рассматривать как некоторую точку, в которой происходит обращение к подпрограмме. В тот момент, когда транзакт входит в блок, соответствующая подпрограмма выполняет свою функцию. Далее транзакт пытается войти в следующий блок. Такое продвижение продолжается до тех пор, пока не наступит одна из следующих возможных ситуаций:

- транзакт входит в блок, функцией которого является задержка транзакта на некоторое время;
- транзакт входит в блок, функцией которого является удаление транзакта из модели;
- транзакт пытается войти в следующий блок, который его не принимает. В этом случае транзакт остается в том блоке, в котором остановился, и становится в очередь. Позднее он будет повторять попытку войти в следующий блок, и одна из таких попыток может оказаться успешной. После этого транзакт сможет продолжать свое перемещение в модели.

Обычно в модели существует большое число транзактов (они могут перемещаться согласно некоторой очередности), каждый из которых система пытается продвинуть как можно дальше. При возникновении ситуации, когда дальнейшее перемещение транзакта невозможно, в модели начинается перемещение другого транзакта. Таким образом, выполнение моделирования в системе продолжается.

Каждый блок имеет номер, который присваивается ему автоматически согласно месту его расположения в модели. Кроме того, он может иметь символическое имя (метку), которое используется для ссылки на этот блок при передаче управления. Блок может иметь до восьми операндов, уточняющих его действие. Для удобства записи они обозначаются символами A, B, C, ..., H. Если операнды блока в явном виде не заданы, интерпретатор GPSS устанавливает их значение по умолчанию.

Кроме операторов блоков, разработчик модели может использовать еще два вида операторов: операторы описания объектов и управляющие операторы. *Операторы описания* позволяют определить переменную, функцию, описать емкость памяти (количество каналов многоканальной СМО), параметры таблицы и пр. *Управляющие операторы (команды)* позволяют управлять процессом моделирования, т. е. начинать прогон модели, останавливать его, выполнять сброс накопленной статистики, просматривать значения переменных и т. д. В системе GPSS World управляющие операторы можно как вносить в текст самой модели, так и задавать через командное меню системы.

2.4. Простейшие операторы GPSS

2.4.1. Введение транзактов в модель

Для ввода транзактов в модель используется блок GENERATE (Создать). Он создает (генерирует) транзакты в модели согласно заданному закону распределения интервала времени между их поступлением. Этот блок может иметь 5 операндов и имеет следующий вид:

GENERATE A,B,C,D,E

Операнды A, B, ..., E указывают характеристики входного потока транзактов (в скобках дано значение по умолчанию):

- A – средний интервал времени между поступлением транзактов (0);
- B – половина поля допуска (размах) интервала равномерного распределения (0);
- C – время создания первого транзакта (0);
- D – ограничитель количества создаваемых транзактов (бесконечность);
- E – приоритет создаваемых транзактов (0). В GPSS возможно всего 128 уровней приоритета, они задаются числами от 0 до 127.

На начальном этапе будем рассматривать только равномерное распределение интервалов между транзактами. Для задания других видов распределений нужно использовать специальные функции, что будет рассмотрено позже.

Для планирования событий в системе GPSS World используется список будущих событий. Когда наступает соответствующее модельное время, берется транзакт из этого списка и вводится в блок GENERATE. Таким образом, приходящие транзакты входят в систему через блок GENERATE. В момент входа транзакта в этот блок планируется время прибытия следующего транзакта и записывается в список будущих событий.

GENERATE 5,3

Генерируются транзакты с равномерным распределением интервала поступления 5 ± 3 . Планирование прихода первого транзакта происходит в момент времени 0 путем розыгрыша случайного значения от 2 до 8.

GENERATE 10

Поскольку размах (операнд В) по умолчанию равен 0, получаем детерминированный, определенный точно, интервал времени между транзактами, равный 10. Первый транзакт приходит в момент времени 10, второй – в момент 20 и т. д.

GENERATE 3,3,10,5

Первый транзакт приходит в момент времени 10. После этого интервалы прибытия находят из равномерного распределения 3 ± 3 , т. е. от 0 до 6. После прихода пяти транзактов работа блока заканчивается.

GENERATE 5,,2

Время создания первого транзакта равно 2. Далее новые транзакты создаются каждые 5 ед. времени, т. е. создание транзактов будет происходить в детерминированные моменты времени: 2, 7, 12, 17, Здесь пропущен операнд В и ему присвоено значение по умолчанию.

GENERATE 8,1,,,4

Транзакты создаются с интервалом 8 ± 1 . Приход первого транзакта рассчитывается в момент времени 0. Количество транзактов не ограничено. Пропущены операнды С и D. Все создаваемые транзакты имеют приоритет, равный 4.

2.4.2. Удаление транзактов из модели

Транзакты удаляются из модели, попадая в блок TERMINATE (Уничтожить). Блоки TERMINATE всегда позволяют выйти из модели всем транзактам, которые пытаются это сделать. Блок TERMINATE может иметь только один операнд, т. е. имеет вид

TERMINATE A

Операнд А (по умолчанию 0) указывает число, на которое уменьшается содержимое счетчика числа завершений.

Счетчик числа завершений – это переменная, которая используется для управления окончанием процесса моделирования. Такой счетчик в модели только один, хотя блоков TERMINATE может быть несколько. При достижении счетчиком завершений значения 0 моделирование завершается. Его начальное значение задается с помощью управляющего оператора (команды) START (Начать моделирование), имеющего формат

START A

Операнд А – начальное значение счетчика завершений (по умолчанию 0).

Команда START может быть записана последней в тексте модели, и тогда моделирование начнется сразу же после успешной компиляции модели. Если в тексте модели нет команды START, то после компиляции модели система ожидает от пользователя дальнейших команд. Команда START может быть введена далее через главное меню.

Счетчик завершений используется для:

- а) ограничения числа транзактов, «прогоняемых» в модели;
- б) ограничения времени, в течение которого исследуется объект.

Например, мы хотим закончить моделирование, когда сотый по счету клиент покинет модель. Тогда можно задать в соответствующем операторе TERMINATE значение операнда А, равное 1:

TERMINATE 1

Начать моделирование можно командой (оператором)

START 100

При удалении каждого транзакта, имитирующего клиента, счетчик числа завершений уменьшается на единицу. Когда сто клиентов пройдут блок TERMINATE, счетчик завершений станет равен нулю, и моделирование прекратится.

Другая ситуация, если мы хотим закончить моделирование по истечении некоторого времени. Например, мы моделируем работу парикмахерской в течение рабочего дня (8 ч), а единицей времени является 1 мин. Тогда в момент времени, равный 480, моделирование должно быть закончено, независимо от того,

сколько клиентов через нее прошло и сколько осталось. Для этого в модель вводится дополнительный сегмент, соответствующий таймеру времени, вида

GENERATE 480
TERMINATE 1

При этом начальное значение счетчика завершений командой START должно быть равно единице, а все другие блоки TERMINATE, уничтожающие транзакты-клиенты, не должны содержать параметров. В момент модельного времени, равный 480, будет сгенерирован транзакт-таймер и сразу же уничтожен. При этом счетчик завершений станет равен нулю, и моделирование прекратится.

2.4.3. Управление устройствами

Объект *устройство* является аналогом канала обслуживания в СМО. В каждый момент времени оно может быть занято только одним транзактом. Если устройство занято, транзакты ожидают в очереди его освобождения.

Занятие устройства транзактом выполняется блоком SEIZE (Занять) формата

SEIZE A

Операнд A – символическое имя или номер устройства.

Правила записи всех символических имен в GPSS (имени устройства, очереди, переменной или метки в модели) следующие:

- первые три символа должны быть буквами;
- имя не должно быть ключевым словом системы GPSS;
- все символы строчных букв преобразуются в прописные.

Вход транзакта в блок SEIZE моделирует занятие устройства. Если транзакт пытается войти в этот блок, то производятся следующие действия:

- проверяется состояние устройства;
- если устройство свободно, то состояние устройства изменяется на *занято*, а транзакту разрешается дальнейшее продвижение;
- если же устройство занято, то транзакту не разрешается вход в блок SEIZE, он остается в предыдущем блоке, и на этом этапе его продвижение в модели заканчивается.

Освобождение устройства выполняется блоком RELEASE (Освободить) формата

RELEASE A

Операнд A – символическое имя или номер устройства.

Этот блок разрешает вход и выход транзакту в любом случае. Когда транзакт входит в этот блок, состояние устройства изменяется на *свободно*.

Если разработчик модели использует пару блоков SEIZE – RELEASE с одним и тем же именем устройства, то тем самым он сообщает системе о наличии такого устройства и необходимости регистрировать статистические данные по нему. Коэффициент использования, количество занимавших устройство транзактов, среднее время обслуживания и другие статистические данные собираются по всем устройствам автоматически и могут быть получены пользователем по окончании моделирования.

2.4.4. Задержка транзактов во времени

Моделирование обслуживания транзакта на устройстве выполняется задержкой этого транзакта на случайное время обслуживания. Реализуется это в GPSS с помощью блока ADVANCE (Задержать) формата

ADVANCE A,B

Операнды блока имеют следующие значения:

A – среднее значение времени задержки (по умолчанию 0);

B – половина поля допуска (размах) интервала времени задержки (0).

Блок ADVANCE никогда не препятствует входу транзакта. Любое число транзактов может находиться в этом блоке одновременно. Когда транзакт попадает в блок ADVANCE, разыгрывается случайное значение времени пребывания в нем, согласно операндам A и B. Транзакт будет там находиться до истечения этого времени. Как и в случае блока GENERATE, для реализации других видов распределений, отличных от равномерного, нужно использовать в качестве аргументов специальные функции.

Примеры

ADVANCE 10,2

Время задержки распределено равномерно в интервале 10 ± 2 , т. е. может принимать значения от 8 до 12.

Время задержки любого транзакта равно 100 – неслучайное значение, так как параметр В по умолчанию равен нулю.

ADVANCE

Фиктивный блок, не задерживающий транзакты.

```
SEIZE      MASTER
ADVANCE    16,4
RELEASE    MASTER
```

Классический пример использования цепочки SEIZE – ADVANCE – RELEASE для моделирования работы устройства (парикмахера из нашего примера). Транзакт, двигающийся по этой цепочке, займет устройство с символическим именем MASTER, задержится там на (16 ± 4) ед. времени, и затем покинет его. Когда данный транзакт войдет в блок RELEASE и процедура этого блока отработает, следующий транзакт сможет использовать устройство MASTER.

2.4.5. Сбор статистики при ожидании в очереди

Как уже отмечалось выше, если устройство занято каким-либо транзактом, то блок SEIZE запрещает вход всем остальным транзактам. Транзакты, которые тоже хотели бы занять устройство, остаются в предыдущем блоке. Там они могут накапливаться, образуя очередь. Дисциплина обслуживания этой очереди, принятая по умолчанию в GPSS, – «первым пришел – первым обслужен в пределах одного класса приоритетов». Таким образом, после освобождения устройства, из очереди выбирается тот транзакт, который имеет больший приоритет, а среди транзактов одинакового приоритета – тот, который пришел ранее. Очереди в системе образуются автоматически в результате того, что транзакты вынуждены ожидать освобождения устройства. По некоторым очередям разработчик модели хочет собирать статистику моделирования, а именно:

- максимальное число транзактов в очереди;
- среднее число транзактов в очереди;
- общее число транзактов, прошедших через очередь;
- число транзактов, которые были зарегистрированы, но не потратили никакого времени на ожидание в очереди (нулевые входы), и их процент от общего числа транзактов;
- среднее время ожидания в очереди (с учетом и без учета нулевых входов).

Для сбора такой статистики (а не для организации очереди) существуют блоки QUEUE (Стать в очередь) и DEPART (Покинуть очередь):

```
QUEUE  A,B
DEPART A,B
```

Операнды данных блоков означают следующее:

А – символическое имя или номер очереди, к которой нужно присоединиться или покинуть (по умолчанию «ошибка», т. е. обязательно должно быть задано);








В – число занимаемых или освобождаемых мест в очереди (по умолчанию 1).

Блоки QUEUE и DEPART всегда пропускают транзакт. При входе транзакта в блок QUEUE моделирующая программа GPSS выполняет действия, фиксирующие вхождение транзакта в очередь (увеличение счетчика входов, запоминание времени постановки транзакта в очередь и т. д.). Когда же транзакт проходит блок DEPART, выполняются действия, регистрирующие выход транзакта из очереди, и расчет соответствующих характеристик этого транзакта. Следует отметить, что моделирующий алгоритм GPSS устроен так, что среднее время пребывания в очереди рассчитывается с учетом и тех транзактов, которые в момент окончания моделирования находятся в очереди. Поэтому среднее время пребывания в очереди получается несколько заниженным, так как транзакты, находящиеся в очереди, должны были бы оставаться в ней еще некоторое время. Статистику по очередям пользователь получает в отчете по окончании моделирования.

В системе GPSS транзакт может присутствовать одновременно в пяти очередях. Ситуация, когда присутствие транзакта необходимо в нескольких очередях одновременно, может возникнуть, например, при моделировании поведения покупателя в магазине, который занимает очереди сразу в несколько отделов.

В табл. 6 показаны фигуры блок-схемы, которые соответствуют выше описанным блокам GPSS.

Таблица 6. Фигуры блок-схемы, соответствующие операторам блоков

GENERATE A,B	
TERMINATE A	
SEIZE A	
RELEASE	
ADVANCE A,B	
QUEUE A	
DEPART A	

2.5. Модель работы парикмахерской

Рассмотрим теперь модель парикмахерской с одним мастером, которая описана в разделе 2.1 (рис. 15). Блок-схема этой модели приведена на рис. 16.

*Модель работы парикмахерской

GENERATE	18,6	;Приход клиентов	(1)
QUEUE	QMASTER	;Присоединение к очереди	(2)
SEIZE	MASTER	;Переход в кресло парикмахера	(3)
DEPART	QMASTER	;Уход из очереди	(4)
ADVANCE	16,4	;Обслуживание у парикмахера	(5)
RELEASE	MASTER	;Освобождение парикмахера	(6)
TERMINATE		;Уход из парикмахерской	(7)
GENERATE	480	;Приход транзакта-таймера	(8)
TERMINATE	1	;Завершение прогона	(9)

Рис. 15. Модель работы парикмахерской с одним мастером



Рис. 16. Блок-схема модели парикмахерской

Текст, расположенный после знаков «*» и «;», является комментарием и системой не обрабатывается. Номера блоков, записанные справа в скобках, не являются частью программы и даны для удобства ссылок на операторы блоков.

Для того, чтобы начать моделирование, должна быть подана команда START 1. Эта команда дает начальное значение 1 счетчику числа завершений и начинает процесс моделирования.

Итак, транзакты создаются блоком GENERATE (1) согласно равномерному распределению с интервалом 18 ± 6 . Транзакт моделирует клиента, пришедшего в парикмахерскую. Затем созданный транзакт продвигается в блок (2), где регистрируется его присоединение к очереди QMASTER. Затем транзакт пытается занять устройство MASTER, имитирующее парикмахера (3). Если ему это удастся, то состояние устройства меняется на *занято*, а транзакт продвигается в блок (4), регистрирующий его уход из очереди QMASTER. В случае неудачи транзакт остается в блоке (2) и ожидает в очереди освобождения устройства. После регистрации выхода из очереди, транзакт поступает в блок (5), где моделируется обслуживание клиента парикмахером задержкой на время 16 ± 4 . Затем транзакт поступает в блок (6), который изменяет состояние устройства на *свободно*, и далее проходит в блок (7), где уничтожается без изменения счетчика завершений. Второй сегмент модели состоит из двух блоков (8) и (9). Блок (8) генерирует транзакт-таймер, который приходит в момент модельного времени, равный 480, и символизирует звонок к окончанию рабочего дня. Этот транзакт тут же уничтожается блоком (9), при этом значение счетчика завершений уменьшается на единицу и становится равным нулю. Моделирование прекращается.

Очевидным недостатком этой модели является то, что моделирование прекращается резко, без всяких подготовительных процедур. Если бы эта ситуация происходила в реальной жизни, то клиент, находящийся в кресле парикмахера, в момент звонка к окончанию дня остался бы недостриженным, а клиенты, ожидающие в очереди, ушли бы, не получив обслуживания.

Таким образом, в модели неявным образом объявлено устройство с именем MASTER. Транзакты, ожидающие в очереди к этому устройству, скапливаются в блоке (2) QUEUE QMASTER. По этой очереди регистрируется статистика под символическим именем QMASTER.

2.6. Логика работы моделирующей системы

Для четкого понимания принципов синхронизации движения транзактов по модели необходимо ввести понятия цепей текущих и будущих событий.

Цепь текущих событий (ЦТС) – это список транзактов, продвижение которых планируется в течение текущего модельного времени или как только будет возможно, если транзакт ожидает освобождения устройства.

Цепь будущих событий (ЦБС) – это список транзактов, продвижение которых планируется начать в некоторый момент времени в будущем. В этот список попадают транзакты, задержанные блоком ADVANCE, и транзакты, которые должны войти в модель через блок GENERATE в некоторый более

поздний момент времени. Транзакты в этом списке отсортированы в порядке запланированных будущих моментов времени.

Любой транзакт одновременно находится и в блоке, и в какой-либо цепи событий. В процессе работы моделирующей программы транзакты могут переноситься из одной цепи событий в другую.

Работу моделирующего алгоритма можно разделить на две основные фазы:

1. Фаза коррекции таймера модельного времени.
2. Фаза просмотра списка текущих событий.

Фаза коррекции таймера модельного времени заключается в том, что в начале отсортированной ЦБС находится первый транзакт, и таймер модельного времени продвигается до значения времени его движения. Первый транзакт переносится из цепи будущих событий в цепь текущих событий. Кроме того, в ЦТС переносятся и другие транзакты из ЦБС, если их время движения совпадает с текущим значением таймера модельного времени.

Фаза просмотра списка текущих событий состоит в выполнении следующих действий:

- Транзакт, находящийся в начале ЦТС, продвигается через возможно большее число блоков.
- Если продвигавшийся транзакт прошел блок SEIZE или RELEASE, то ЦТС должна просматриваться сначала. Причина этого в том, что при прохождении блока RELEASE могут создаться условия продвижения ранее заблокированных из-за занятости устройства транзактов. Необходимость нового просмотра ЦТС в результате выполнения блока SEIZE в данном пособии не рассматривается.
- Если ранее продвигавшийся транзакт не проходил блоков SEIZE и RELEASE, то ЦТС продолжает просматриваться далее до тех пор, пока все возможности продвижения транзактов в ЦТС не будут исчерпаны.

Каждый транзакт в ЦТС или ЦБС представляется в виде 5-позиционной записи следующего вида:

[Номер транзакта, Приоритет, Время движения, Номер текущего блока, Номер следующего блока].

Например: [9,0,68,5,6]. Транзакт номер 9, имеющий приоритет 0, в момент времени 68 пытается выйти из пятого блока и войти в шестой.

Рассмотрим *пример* состояния цепей событий для тех числовых данных, для которых была построена временная диаграмма событий в разделе 2.2. Разыгранные значения интервалов времени прихода транзактов составили 14, 13, 17, 15, ..., разыгранные значения времени обслуживания клиента составили 18, 12, 14,

Как уже было показано, события в системе происходят в моменты времени 14, 27, 32, 44, 58, 59, Таймер модельного времени будет изменяться именно в эти моменты времени. В табл. 7 каждому моменту времени соответствуют две строки: первая строка показывает состояние цепей после фазы коррекции таймера, а вторая – после фазы просмотра. При этом показаны только последние состояния цепей в конце каждой фазы, а промежуточное перемещение транзактов не отражено. Что касается нулевого момента времени, то показано состояние цепей после создания первых транзактов, которое выполняется в начале моделирования.

Таблица 7. Состояние цепей текущих и будущих событий

Но- мер стро- ки	Мо- дель- ное вре- мя	Цепь текущих событий ← к началу цепи	Цепь будущих событий ← к началу цепи
1	0	Пусто	[1,0,14,нет,1] [2,0,480,нет,8]
2	14	[1,0,КМР,нет,1]	[2,0,480,нет,8]
3	14	пусто	[3,0,27,нет,1] [1,0,32,5,6] [2,0,480,нет,8]
4	27	[3,0,КМР,нет,1]	[1,0,32,5,6] [2,0,480,нет,8]
5	27	[3,0,КМР,2,3]	[1,0,32,5,6] [4,0,44,нет,1] [2,0,480,нет,8]
6	32	[3,0,КМР,2,3] [1,0,КМР,5,6]	[4,0,44,нет,1] [2,0,480,нет,8]
7	32	Пусто	[4,0,44,нет,1] [3,0,44,5,6] [2,0,480,нет,8]
8	44	[4,0,КМР,нет,1] [3,0,КМР,5,6]	[2,0,480,нет,8]
9	44	пусто	[4,0,58,5,6] [5,0,59,нет,1] [2,0,480,нет,8]
10	58	[4,0,КМР,5,6]	[5,0,59,нет,1] [2,0,480,нет,8]
11	58	пусто	[5,0,59,нет,1] [2,0,480,нет,8]

Поясним коротко состояния цепей, соответствующие каждой строке в табл. 7.

1. В нулевой момент времени создаются два транзакта. Первый символизирует клиента, который придет в парикмахерскую в момент времени, равный 14, а второй – таймер окончания рабочего дня, который придет в момент 480. Цепь текущих событий пуста.

2. В момент времени 14 первый транзакт переносится в цепь текущих событий. Время транзакта в цепи текущих событий будем обозначать КМР (как можно раньше).

3. В момент входа первого транзакта в блок GENERATE планируется приход третьего транзакта-клиента в момент времени $14 + 13 = 27$. Этот транзакт записывается в цепь будущих событий. Первый же транзакт продвигается до тех пор, пока не будет задержан в блоке ADVANCE до момента $14 + 18 = 32$. Тогда он тоже перемещается в цепь будущих событий. Таким образом после фазы просмотра в момент времени 14 все транзакты оказываются в цепи будущих событий, а цепь текущих событий пуста.

4. В момент времени 27 третий транзакт переносится в ЦТС.

5. В момент входа третьего транзакта в блок GENERATE планируется приход следующего транзакта-клиента в момент $27 + 17 = 44$, который заносится под номером 4 в ЦБС. Третий же транзакт задерживается в блоке номер 2 (QUEUE), поскольку устройство занято и он не может войти в следующий блок под номером 3 (SEIZE). Третий транзакт остается в ЦТС (стоит в очереди) и при первой возможности продвигается далее.

6. В момент 32 происходит событие *Окончание обслуживания транзакта 1*, и этот транзакт переносится в конец ЦТС.

7. Первый транзакт проходит в блок 6 (RELEASE), освобождая устройство, а затем уничтожается блоком TERMINATE. Поскольку этот транзакт прошел через блок RELEASE, то ЦТС должна просматриваться сначала. Теперь третий транзакт имеет возможность двигаться, он занимает устройство и задерживается в блоке ADVANCE до момента $32 + 12 = 44$. Третий транзакт помещается в цепи будущих событий за четвертым, как позже попавший в ЦБС.

8. В момент времени 44 в цепь текущих событий перемещаются два транзакта: четвертый, который поступает в этот момент времени; и третий, который должен закончить обслуживание.

9. Четвертый транзакт стоит в списке первым, он и начинает движение. При входе его в блок GENERATE планируется приход следующего транзакта в момент $44 + 15 = 59$. Этот транзакт номер пять записывается в ЦБС. Затем движение четвертого транзакта прекращается в блоке номер 2, поскольку устройство еще занято третьим транзактом. Потом система пытается продвинуть третий транзакт. Он заканчивает обслуживание на устройстве и уничтожается системой. Поскольку третий транзакт прошел блок RELEASE, система опять просматривает ЦТС, в результате четвертый транзакт получает возможность движения и занимает устройство до момента $44 + 14 = 58$, т. е. переносится в ЦБС. Цепь текущих событий остается пуста.

10. Четвертый транзакт переносится в ЦТС.

11. Четвертый транзакт заканчивает обслуживание и уничтожается в блоке TERMINATE.

Таким образом моделирование будет продолжено и дальше. Следующее событие произойдет в момент 59.

2.7. Расчет статистических характеристик объектов

В системе GPSS автоматически собирается статистика по всем устройствам и очередям, для которых заданы блоки QUEUE –DEPART. Для лучшего понимания сути этих статистических характеристик, рассмотрим пример ручного расчета по устройству MASTER и очереди QMASTER в модели парикмахерской из раздела 2.5. Будем использовать исходные данные, приведенные в разделе 2.1, а именно: интервалы времени между приходами транзактов 14, 13, 17, 15, ... и значения времени обслуживания клиентов 18, 12, 14, ... Для данного примера была построена временная диаграмма (см. рис. 14). Допустим, моделирование прервано в момент времени 60.

Характеристики очереди QMASTER:

- Общее число входов в очередь равно 4. Каждый транзакт, поступающий в систему, проходит блок QUEUE, т. е. регистрируется в очереди. К моменту модельного времени 60 четыре транзакта пройдут этот блок.

- Число нулевых входов равно 3. Транзакты, которым не пришлось ожидать в очереди (в момент их прихода устройство было свободно), регистрируют выход из очереди через блок DEPART в тот же момент, что и вход в блок QUEUE. Такие транзакты называются нулевыми входами в очередь. В нашем примере из четырех зарегистрированных в очереди транзактов первый, третий и четвертый оказались нулевыми входами.

- Максимальное количество транзактов в очереди равно 1. Следует отметить, что в системе GPSS значение этой величины никогда не будет меньше единицы. Даже тот транзакт, который не стоял фактически в очереди, в ней зарегистрирован. Если представить ситуацию, что ни один транзакт не ожидал в очереди (все они были нулевыми входами), система все равно даст значение этого показателя, равное единице.

- Среднее значение количества транзактов в очереди равно 0,083. Для расчета этой величины нарисую ось модельного времени до момента окончания моделирования (60) и отметим на ней количество транзактов, находившихся в очереди в течение каждого интервала времени (рис. 17). Найдем сумму всех временных интервалов, во время которых наблюдалось одно и то же значение количества транзактов в очереди.

Ноль транзактов: $(14 - 0) + (27 - 14) + (44 - 32) + (58 - 44) + (59 - 58) + (60 - 59) = 55$.

Один транзакт: $(32 - 27) = 5$.

Чтобы найти среднее число транзактов в очереди, нужно каждое значение количества ожидающих транзактов умножить на долю времени, в течение которого наблюдалось это количество транзактов в очереди:

$$L = 0 \cdot \frac{55}{60} + 1 \cdot \frac{5}{60} = 0,083.$$

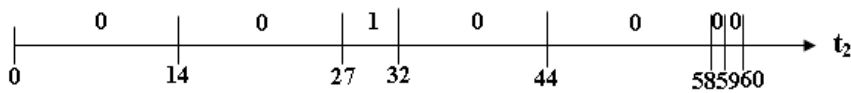


Рис. 17. Количество транзактов в очереди в каждый момент времени

- Среднее время пребывания в очереди равно 1,25. Найдем сумму времен ожидания каждого транзакта и разделим ее на количество транзактов:

$$t_{ож} = (0 + 5 + 0 + 0) : 4 = 1,25.$$

- Среднее время пребывания в очереди без учета нулевых входов равно 5. При его расчете не учитываются транзакты, которым не пришлось ожидать:

$$t_{ож} = 5 : 1 = 5.$$

Характеристики устройства MASTER:

- Число обслуживаний равно 4. Фактически число обслуживаний есть количество транзактов, входивших в блок SEIZE. Таким образом учитывается и тот транзакт, который находится на устройстве в момент окончания моделирования.

- Коэффициент использования устройства равен 0,75. Эта величина есть отношение времени, в течение которого устройство было занято обслуживанием, к общему модельному времени. Общее модельное время равно 60 ед. Устройство было занято на интервалах от 14 до 58 и от 59 до 60, т. е. $(58 - 14) + (60 - 59) = 45$

ед. времени. Коэффициент использования устройства $K = \frac{45}{60} = 0,75$.

- Средняя продолжительность одного обслуживания равна 11,25 ед. модельного времени. Разделим время, в течение которого устройство было занято обслуживанием, на число обслуживаний и получим

$t_{об} = \frac{45}{4} = 11,25$. Таким образом, в расчете участвует и тот транзакт, который в данный момент находится на устройстве, хотя его обслуживание только начато. Правильнее было бы рассчитать время обслуживания первых трех транзактов и разделить на три: $\frac{18+12+14}{3} = 14,67$. Однако в GPSS используется именно

первый способ расчета и полагается, что при большом числе транзактов это занижение времени обслуживания не будет иметь значения.

2.8. Задания для самостоятельной работы

Рассматривается модель работы парикмахерской с одним мастером (см. рис. 15). Значения разыгранных интервалов времени между поступлениями транзактов и временем обслуживания, а также продолжительность моделирования приведены в вариантах заданий. Выполните следующее:

- постройте временную диаграмму событий в системе аналогично рис. 14;
- поясните содержание событий аналогично табл. 4;
- рассчитайте основные характеристики устройства MASTER и очереди QMASTER;
- составьте таблицу состояния цепей текущих и будущих событий аналогично табл. 7 и поясните ее.

Работа должна быть оформлена в тетради для лабораторных работ.

Вариант 1

Интервалы времени между поступлениями транзактов в систему: 5, 11, 3, 8, 10, 12, 4. Интервалы времени обслуживания транзактов: 7, 13, 7, 10, 8. Предельное значение модельного времени 50.

Ответы:

Общее число входов в очередь – 6.

Число нулевых входов – 2.

Максимальное количество транзактов в очереди – 2.

Среднее количество транзактов в очереди – 0,58.

Среднее время пребывания в очереди – 4,83.

Среднее время пребывания в очереди без учета нулевых входов – 7,25.

Число обслуживаний – 5.

Коэффициент использования устройства – 0,82.

Средняя продолжительность обслуживания – 8,2.

Вариант 2

Интервалы времени между поступлениями транзактов в систему: 8, 5, 10, 4, 6, 8, 10, 11. Интервалы времени обслуживания транзактов: 10, 14, 15, 8, 13. Предельное значение модельного времени 60.

Ответы:

Общее число входов в очередь – 7.

Число нулевых входов – 1.

Максимальное количество транзактов в очереди – 3.

Среднее количество транзактов в очереди – 1,4.

Среднее время пребывания в очереди – 12.

Среднее время пребывания в очереди без учета нулевых входов – 14.

Число обслуживаний – 5.

Коэффициент использования устройства – 0,87.

Средняя продолжительность обслуживания – 10,4.

Вариант 3

Интервалы времени между поступлениями транзактов в систему: 3, 14, 6, 8, 7, 12, 14. Интервалы времени обслуживания транзактов: 10, 5, 12, 6, 10, 7, 8. Предельное значение модельного времени 55.

Ответы:

Общее число входов в очередь – 6.

Число нулевых входов – 3.

Максимальное количество транзактов в очереди – 1.

Среднее количество транзактов в очереди – 0,15.

Среднее время пребывания в очереди – 1,33.

Среднее время пребывания в очереди без учета нулевых входов – 2,67.

Число обслуживаний – 6.

Коэффициент использования устройства – 0,85.

Средняя продолжительность обслуживания – 7,83.

Вариант 4

Интервалы времени между поступлениями транзактов в систему: 10, 5, 4, 8, 2, 6, 4, 9, 6. Интервалы времени обслуживания транзактов: 5, 9, 15, 6, 4, 7. Предельное значение модельного времени 50.

Ответы:

Общее число входов в очередь – 8.

Число нулевых входов – 2.

Максимальное количество транзактов в очереди – 3.

Среднее количество транзактов в очереди – 1,2.

Среднее время пребывания в очереди – 7,5.

Среднее время пребывания в очереди без учета нулевых входов – 10.

Число обслуживаний – 6.

Коэффициент использования устройства – 0,8.

Средняя продолжительность обслуживания – 6,67.

Вариант 5

Интервалы времени между поступлениями транзактов в систему: 5, 20, 15, 30, 6, 8, 15, 10. Интервалы времени обслуживания транзактов: 25, 30, 10, 10, 15, 10. Предельное значение модельного времени 100.

Ответы:

Общее число входов в очередь – 7.

Число нулевых входов – 2.

Максимальное количество транзактов в очереди – 1.

Среднее количество транзактов в очереди – 0,41.

Среднее время пребывания в очереди – 5,86.

Среднее время пребывания в очереди без учета нулевых входов – 8,2.

Число обслуживаний – 6.

Коэффициент использования устройства – 0,95.

Средняя продолжительность обслуживания – 15,8.

Вариант 6

Интервалы времени между поступлениями транзактов в систему: 1, 8, 20, 40, 6, 8, 14, 20. Интервалы времени обслуживания транзактов: 20, 15, 20, 20, 10, 5, 8. Предельное значение модельного времени 110.

Ответы:

Общее число входов в очередь – 7.

Число нулевых входов – 2.

Максимальное количество транзактов в очереди – 2.

Среднее количество транзактов в очереди – 0,51.

Среднее время пребывания в очереди – 8.
Среднее время пребывания в очереди без учета нулевых входов – 11,2.
Число обслуживаний – 7.
Коэффициент использования устройства – 0,87.
Средняя продолжительность обслуживания – 13,7.

Вариант 7

Интервалы времени между поступлениями транзактов в систему: 6, 12, 4, 9, 11, 13, 8. Интервалы времени обслуживания транзактов: 8, 14, 8, 11, 9, 5. Предельное значение модельного времени 60.

Ответы:

Общее число входов в очередь – 6.
Число нулевых входов – 2.
Максимальное количество транзактов в очереди – 2.
Среднее количество транзактов в очереди – 0,55.
Среднее время пребывания в очереди – 5,5.
Среднее время пребывания в очереди без учета нулевых входов – 8,25.
Число обслуживаний – 5.
Коэффициент использования устройства – 0,83.
Средняя продолжительность обслуживания – 10.

Вариант 8

Интервалы времени между поступлениями транзактов в систему: 4, 10, 2, 7, 9, 11, 3, 8. Интервалы времени обслуживания транзактов: 6, 12, 6, 9, 7, 10. Предельное значение модельного времени 50.

Ответы:

Общее число входов в очередь – 7.
Число нулевых входов – 2.
Максимальное количество транзактов в очереди – 2.
Среднее количество транзактов в очереди – 0,74.
Среднее время пребывания в очереди – 5,29.
Среднее время пребывания в очереди без учета нулевых входов – 7,4.
Число обслуживаний – 6.
Коэффициент использования устройства – 0,84.
Средняя продолжительность обслуживания – 7.

Вариант 9

Интервалы времени между поступлениями транзактов в систему: 5, 5, 15, 15, 5, 10, 7. Интервалы времени обслуживания транзактов: 10, 8, 15, 10, 7, 5, 5. Предельное значение модельного времени 60.

Ответы:

Общее число входов в очередь – 6.
Число нулевых входов – 3.
Максимальное количество транзактов в очереди – 1.
Среднее количество транзактов в очереди – 0,2.
Среднее время пребывания в очереди – 2.
Среднее время пребывания в очереди без учета нулевых входов – 4.
Число обслуживаний – 6.
Коэффициент использования устройства – 0,88.
Средняя продолжительность обслуживания – 8,83.

Вариант 10

Интервалы времени между поступлениями транзактов в систему: 1, 5, 7, 10, 2, 8, 6, 10, 4. Интервалы времени обслуживания транзактов: 5, 4, 6, 6, 10, 5, 8, 5. Предельное значение модельного времени 50.

Ответы:

Общее число входов в очередь – 8.
Число нулевых входов – 4.
Максимальное количество транзактов в очереди – 1.
Среднее количество транзактов в очереди – 0,32.
Среднее время пребывания в очереди – 2,5.
Среднее время пребывания в очереди без учета нулевых входов – 5.
Число обслуживаний – 7.
Коэффициент использования устройства – 0,84.
Средняя продолжительность обслуживания – 6.

Контрольные вопросы

1. Что такое таймер модельного времени? Какие значения он может принимать?
2. Что такое событие? Какие категории событий можно выделить? Приведите примеры событий.
3. Как планируется событие «приход клиента» в модели парикмахерской? Проиллюстрируйте это временной диаграммой.
4. Как планируется событие «окончание обслуживания» в модели парикмахерской? Проиллюстрируйте это временной диаграммой.
5. В чем суть метода переменного приращения таймера модельного времени?
6. Что такое транзакт?
7. На какие виды подразделяются объекты статической категории (оборудования)? Для чего они применяются?
8. Какие статистические объекты имеются в GPSS? Для чего они применяются?
9. В чем отличие очереди в системе массового обслуживания и очереди как объекта системы GPSS?
10. Из каких объектов строится модель на языке GPSS? Как происходит процесс моделирования?
11. Какие виды операторов имеются в GPSS?
12. Какой блок используется для создания транзактов?
13. Как задать детерминированный интервал времени между поступлениями транзактов в систему?
14. Как ограничить число создаваемых транзактов?
15. Что такое счетчик числа завершений? Как и для чего он используется?
16. Каковы правила записи символических имен в GPSS?
17. Какие из изученных в данной работе блоков всегда разрешают выход транзакта, а какие могут давать на это запрет?
18. Как система GPSS узнает о наличии устройств в модели? По всем ли устройствам собирается статистика?
19. Какие блоки используются для управления работой устройства?
20. Как организуются очереди в модели? По всем ли очередям собирается статистика?
21. Может ли транзакт находиться одновременно в нескольких очередях или на нескольких устройствах?
22. Приведите модель работы парикмахерской и поясните принцип ее работы.
23. Что такое цепи текущих и будущих событий?
24. На какие основные фазы можно разделить работу моделирующего алгоритма? В чем заключается их суть?
25. Как представлен каждый транзакт в цепи текущих или будущих событий?
26. Что означает аббревиатура KMP для обозначения времени в цепи текущих событий?

ЛАБОРАТОРНАЯ РАБОТА 3 ТЕХНОЛОГИЯ РАБОТЫ В СИСТЕМЕ GPSS WORLD

3.1. Общие сведения о системе GPSS World

Система GPSS предназначена для разработки и использования имитационных моделей дискретных сложных систем. Она основана на транзактном способе имитации и больше подходит для моделирования систем массового обслуживания.

В настоящее время используется система GPSS World, разработанная фирмой «Minuteman Software». В данном пособии рассматривается студенческая версия GPSS World. Она бесплатно распространяется через Интернет (www.minutemansoftware.com) и отличается от коммерческой только количественным ограничением – до 150 блоков в составе модели. Эта система имеет стандартный оконный интерфейс и средства для реализации всех этапов имитационного моделирования на языке GPSS:

- текстовый редактор и встроенные шаблоны блоков для ввода модели;
- интерпретирующий алгоритм для синхронизации работы различных компонентов модели и управления таймером модельного времени;
- средства автоматического сбора статистических данных;
- средства отладки модели и контроля динамики процесса имитации;
- средства визуализации процесса моделирования и представления результатов в виде таблиц и графиков.

Запустить систему GPSS World, которая установлена в компьютерном классе, можно командой **Пуск/Программы/GPSS Word Student Version**.

- Выполните запуск системы моделирования GPSS World.*

* Значком треугольника будут обозначены задания, которые требуется *выполнить*, в отличие от остального текста, который нужно *прочитать и понять*.

3.2. Работа с файлами

Пункт главного меню *File* служит для работы с файлами документов. В системе GPSS World используются несколько типов файлов:

***.gps** – текст имитационной модели на языке моделирования GPSS. Чтобы осуществить прогон этой модели, ее нужно откомпилировать и запустить на исполнение.

***.txt** – обычный текстовый файл, который можно прочитать и модифицировать при помощи текстового редактора.

***.gpr** – отчет о результатах проведенного моделирования. В этом файле сохраняется содержимое окна *Отчет* (REPORT).

***.sim** – сообщения, появившиеся в процессе моделирования системы в окне *Журнал* (JOURNAL).

Основные разделы пункта меню *File*:

New – создать новый файл (при выборе этого раздела появляется диалоговое окно, которое позволяет выбрать тип создаваемого файла: пункт *Model* создает файл с текстом имитационной модели на языке GPSS, который имеет расширение *.gps, а пункт *Text File* приводит к созданию обычного текстового файла с расширением *.txt);

Open – открыть уже существующий файл (в папке *Sample Models* содержится большой набор примеров моделей, которые можно использовать как основу для своих разработок);

Close – закрыть файл;

Save – сохранить файл;

Save As – сохранить как (под другим именем или в другой папке);

Print – печать;

Internet;

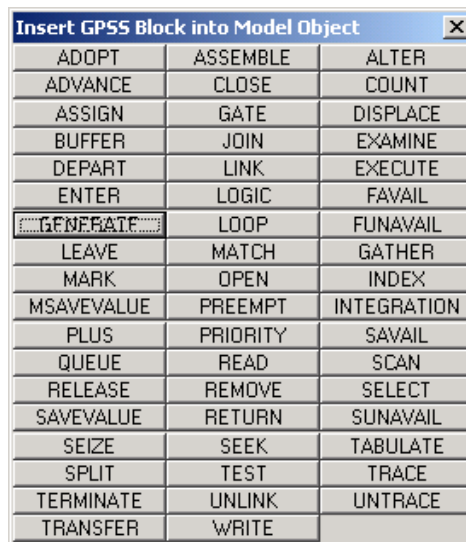
Recent File – последний файл;

Exit – выход из программы.

3.3. Ввод текста модели

► Задайте команду **File/New/Model/OK**. В появившемся окне *Untitled Model 1* (Безымянная модель 1) вручную можно ввести текст модели, используя встроенный текстовый редактор, а можно использовать команду **Edit/Insert GPSS Blocks** (*Правка/Вставить блоки GPSS*). В последнем случае необходимо выбрать конкретный блок (рис. 18), а затем задать его параметры (рис. 19).

Блоки модели лучше располагать один под другим, начиная со второй позиции табуляции, а параметры блоков – с третьей. С первой позиции табуляции располагаются имена блоков (метки). Комментарии могут начинаться со знака «*» в начале строки и со знака «;» в середине строки.



ADOPT	ASSEMBLE	ALTER
ADVANCE	CLOSE	COUNT
ASSIGN	GATE	DISPLACE
BUFFER	JOIN	EXAMINE
DEPART	LINK	EXECUTE
ENTER	LOGIC	FAVAIL
GENERATE	LOOP	FUNAVAIL
LEAVE	MATCH	GATHER
MARK	OPEN	INDEX
MSAVEVALUE	PREEMPT	INTEGRATION
PLUS	PRIORITY	SAVAIL
QUEUE	READ	SCAN
RELEASE	REMOVE	SELECT
SAVEVALUE	RETURN	SUNAVAIL
SEIZE	SEEK	TABULATE
SPLIT	TEST	TRACE
TERMINATE	UNLINK	UNTRACE
TRANSFER	WRITE	

Рис. 18. Окно вставки блока

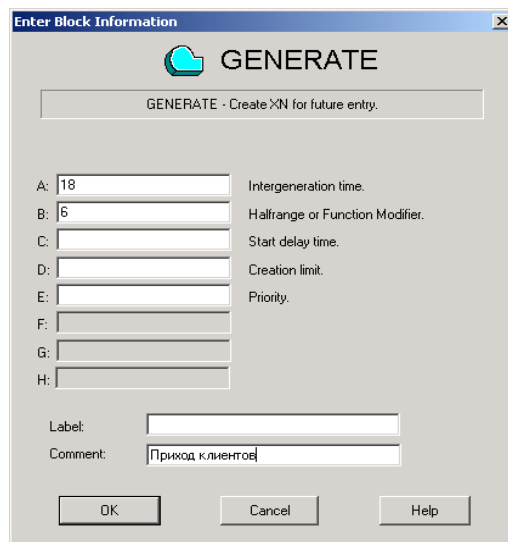


Рис. 19. Окно задания параметров блока

► Введите текст модели, как показано на рис. 20. Сохраните его под именем *Lab2.gps* в своей личной папке.

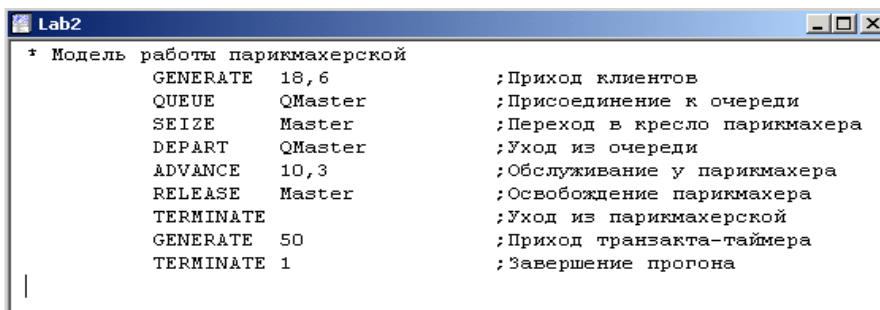


Рис. 20. Окно исходной модели

3.4. Компиляция и исправление ошибок

► Для начала компиляции необходимо, чтобы окно модели было активным. Для этого щелкните мышью в любом месте окна модели. Выполните команду **Command/Create Simulation** (*Команда/Создать имитацию*). Можно использовать также «горячие клавиши» *Ctrl+Alt+S*. Начинается процесс компиляции модели и автоматически открывается окно *Журнал* (JOURNAL). В нем записываются все взаимодействия пользователя с системой и сообщения GPSS World, в частности информация о найденных синтаксических ошибках. Если компиляция прошла без ошибок и в тексте модели есть управляющая команда **START**, то немедленно будет начато выполнение программы имитации, т. е. прогон модели. В нашем случае такой команды нет, поэтому после компиляции в окне *Журнал* будет напечатано сообщение о готовности с датой и временем окончания компиляции (рис. 21), и система будет ожидать дальнейших команд пользователя.

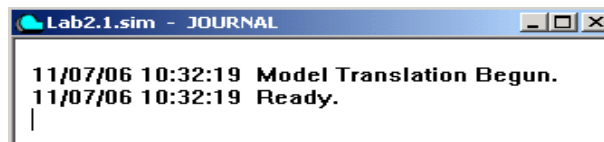


Рис. 21. Окно Журнал

► Рассмотрим исправление ошибок компиляции. Для этого перейдите в окно модели, используя пункт главного меню *Window*, и внесите в текст программы две ошибки: первый оператор исправьте на **TENER-ATE 18,6**, а оператор задержки напишите в виде **ADANCE 10,3**. Вновь откомпилируйте модель командой **Command/Retranslate** (*Перекомпилировать*). В окне *Журнал* появятся сообщения об ошибках компиляции (рис. 22). Для каждой ошибки указывается номер строки и столбца, в котором обнаружен неверный символ (Line 2 Col 11) и тип ошибки (Invalid number. Expecting a GPSS Verb (*Неверный символ. Ожидается оператор GPSS*)). Далее приводится строка программы, в которой обнаружена ошибка. Еще ниже дается сообщение: **Model Translation Aborted** (*Трансляция модели прервана*).

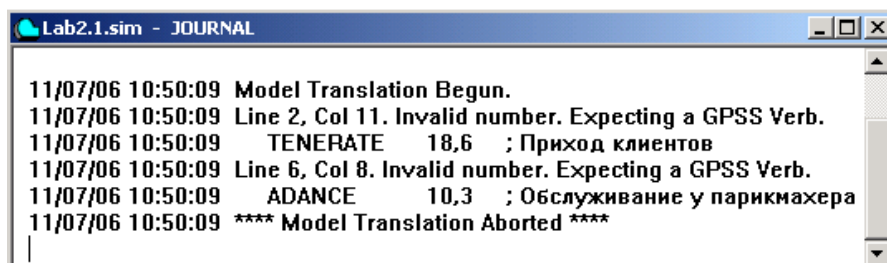


Рис. 22. Окно Журнал с ошибками компиляции

Чтобы быстро найти строку, в которой сделана ошибка, используется пункт главного меню *Search* (Поиск). В нем содержатся команды:

- *Go to Line...* (*Ctrl+Alt+G*) – перейти к строке (далее в диалоговом окне указывается номер строки).
- *Next Error* (*Ctrl+Alt+N*) – следующая ошибка.
- *Previous Error* (*Ctrl+Alt+P*) – предыдущая ошибка.

Все ошибки времени компиляции образуют циклический список, это значит, что команда *Next Error* с последней ошибки переводит на первую.

► Используя команды поиска ошибок из пункта меню *Search*, исправьте неверные операторы и перетранслируйте модель.

3.5. Выполнение программы имитации

Однократное выполнение программы имитационной модели будем называть *прогоном модели*. Прогон начинается, когда система встречает управляющий оператор **START**, который может быть в тексте модели, также его можно задать в интерактивном режиме командой **Command/START**. Появляется диалоговое окно, в котором следует уточнить значение счетчика числа завершений. В нашем примере моделирование должно завершаться с приходом транзакта-таймера, который уменьшает счетчик завершений на единицу (см. последнюю строку программы). Поэтому начальное значение счетчика завершений должно быть равно 1, т. е. команда начала моделирования имеет вид **START 1**.

В ходе выполнения моделирования текущие сообщения выводятся в журнал и отражаются в строке состояния. В случае успешного окончания прогона модели, в журнале указывается время окончания моделирования и сообщение о формировании отчета. Отчет о результатах моделирования формируется автоматически в окне *Отчет* (REPORT).

► Выполните команду **START 1** и изучите сообщения в окне *Журнал*. Моделирование закончилось в 50 мин модельного времени (Clock is 50,000 000). Это же время указано в столбце END TIME окна *Отчет*.

Повторные команды **START**, или повторные прогоны, выполняются с текущего значения модельного времени. Учитывается также и накопленная статистика моделирования, т. е. моделирование продолжается с того состояния, в котором модель находилась в момент прекращения предыдущего прогона. Сбросить модельное время и статистику моделирования до нуля можно командой **Command/CLEAR**.

► Выполните еще раз команду **START 1**. Будет сформирован новый отчет, в котором значение END TIME равно 100 и статистика моделирования учитывает информацию, накопленную во время обоих прогонов. Теперь задайте команду **Command/CLEAR** и выполните еще один прогон модели. Время окончания моделирования будет равно 50, т. е. моделирование было начато сначала, с нулевого момента времени, когда в системе не было еще ни одного транзакта.

Иногда не требуется получать отчет о результатах моделирования при каждом прогоне модели. Отменить его формирование можно, если в команде **START** задать параметр NP (не формировать отчет):

START 1,NP

(между запятой и параметром NP не должно быть пробела).

Содержимое окна *Журнал* можно сохранить в файле с расширением *.sim. Если закрыть окно *Журнал* без сохранения его содержимого, то объект *имитация*, т. е. откомпилированная программа модели, удаляется из памяти компьютера. В дальнейшем потребуются вновь компиляция модели и запуск имитации на выполнение.

► Закройте без сохранения окно *Журнал* и все созданные ранее отчеты.

Рассмотрим пример ошибок, возникающих в ходе имитации. При появлении любой такой ошибки моделирование аварийно прекращается.

► Внесите в текст программы следующее исправление: в блоке DEPART измените имя очереди на QMast. Выполните компиляцию модели. Такая ошибка не будет выявлена в процессе компиляции. Начните выполнение прогона модели командой **START 1**. Имитация будет остановлена, когда первый транзакт попытается войти в блок DEPART. Система выдаст сообщение, что зафиксирована попытка сделать содержимое очереди отрицательным (рис. 23). Дело в том, что система считает, что имеется две разные очереди: QMaster и QMast. Транзакт был учтен, как поступивший в очередь QMaster (соответствующий счет-

чик количества транзактов в этой очереди был увеличен), а покинуть он пытается другую очередь – QMast. Счетчик количества транзактов должен быть уменьшен на единицу, но тогда он станет отрицательным, ведь его начальное значение равно нулю. Так обнаруживается ошибка в модели, когда имена очереди в блоках QUEUE и DEPART отличаются.

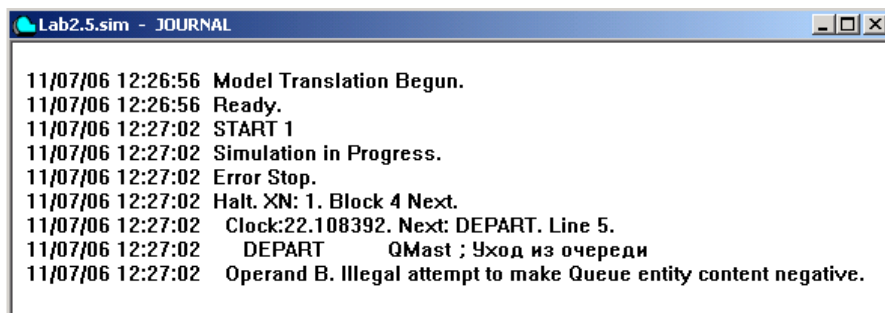


Рис. 23. Окно Журнал с ошибкой времени выполнения

► Исправьте ошибку и перекомпилируйте модель. Выполните прогон модели с формированием отчета.

В нашем примере прогон продолжается ровно 50 ед. модельного времени и затем с приходом транзакта-таймера моделирование прекращается. Напомним (см. лабораторную работу 2), что для моделирования прохождения определенного количества транзактов (например, 100), нужно изменить программу модели следующим образом:

- удалить две последние строки модели, касающиеся транзакта-таймера;
- изменить блок TERMINATE для транзактов, изображающих клиентов, задав ему параметр 1:

TERMINATE 1

- начать моделирование командой START 100.

В этом случае прохождение каждого транзакта через блок TERMINATE уменьшает счетчик завершений на единицу. Поэтому, когда 100 клиентов покинут парикмахерскую, счетчик завершений станет равным нулю и моделирование прекратится.

3.6. Структура стандартного отчета

Содержимое окна *Отчет* показано на рисунках 24–26. В заголовке отчета (рис. 24) указывается имя модели и номер (версия) отчета, а в следующей строке – дата и время его получения. Затем идет общая информация о модели и ее прогоне.

START TIME – модельное время начала сбора статистики; если не использовалась команда RESET, оно равно нулю (подробнее команда RESET будет рассмотрена в следующих лабораторных работах);

END TIME – модельное время окончания сбора статистики (время окончания последнего прогона);

BLOCKS – количество блоков в модели;

FACILITIES – количество устройств в модели;

STORAGES – количество памяти в модели, т. е. объектов GPSS, которые моделируют многоканальные обслуживающие устройства.

Далее в отчете идет таблица символических имен, используемых в модели (рис. 24). Для обозначения устройств, очередей, меток, переменных и т. п. в языке GPSS можно использовать числовые константы и символические имена. Однако в процессе компиляции модели система все символические имена заменяет числами. В столбце NAME указываются символические имена, а в столбце VALUE – соответствующие числовые эквиваленты, присвоенные компилятором.

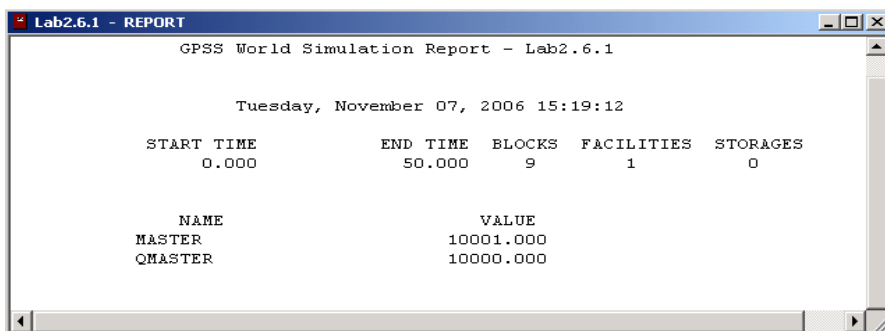


Рис. 24. Содержимое окна Отчет (начало)

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	2	0	0
	2	QUEUE	2	0	0
	3	SEIZE	2	0	0
	4	DEPART	2	0	0
	5	ADVANCE	2	1	0
	6	RELEASE	1	0	0
	7	TERMINATE	1	0	0
	8	GENERATE	1	0	0
	9	TERMINATE	1	0	0

Рис. 25. Содержимое окна *Отчет* (продолжение)

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
MASTER	2	0.377	9.422	1	3	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
QMASTER	1	0	2	2	0.000	0.000	0.000

FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
3	0		53.222	3	5	6		
4	0		58.567	4	0	1		
5	0		100.000	5	0	8		

Рис. 26. Содержимое окна *Отчет* (окончание)

На рис. 25 показан список блоков модели. В столбце ENTRY COUNT приведено общее число транзактов, входивших в данный блок; в столбце CURRENT COUNT – текущее количество транзактов в блоке (на момент окончания моделирования); а в столбце RETRY – количество транзактов, ожидающих готовности для входа в блок. Таким образом, за прошедшее время моделирования (50 ед.) только два транзакта-клиента вошли в систему через блок GENERATE, один из которых покинул парикмахерскую (прошел блок TERMINATE), а другой в момент окончания моделирования находился на обслуживании у парикмахера (блок ADVANCE).

На рис. 26 показано окончание отчета о результатах моделирования, где приведены статистические данные о работе устройства MASTER и очереди QMASTER, а также распечатка FEC (Future Event Chain) – цепи будущих событий. В табл. 8 приведены значения элементов в колонках таблицы устройств (FACILITY), а в табл. 9 – значения для таблицы очередей (QUEUE).

Таблица 8. Значения элементов в колонках таблицы устройств

FACILITY	Имя устройства (символическое или числовое)
ENTRIES	Число входов, т. е. количество транзактов, занимавших устройство
UTIL.	Коэффициент использования устройства, т. е. доля времени, в течение которого устройство было занято обслуживанием
AVE. TIME	Среднее время обслуживания
AVAIL.	Доступность устройства в конце моделирования: 1 означает доступность, 0 – недоступность
OWNER	Номер транзакта, который занимает устройство в момент окончания моделирования; 0 означает, что устройство свободно
PEND	Количество транзактов, которые ожидают прерывания работы устройства (транзакты в цепи Pending Chain)
INTER	Количество прерванных транзактов (в цепи Interrupt Chain)
RETRY	Количество транзактов, ожидающих наступления каких-либо условий для занятия устройства (транзакты в цепи Retry Chain)
DELAY	Количество транзактов, ожидающих возможности занятия устройства (в цепи Delay Chain)

Таблица 9. Значения элементов в колонках таблицы очередей

QUEUE	Имя очереди (символическое или числовое)
MAX	Максимальное количество транзактов в очереди
CONT.	Текущее количество транзактов в очереди (на конец моделирования)
ENTRY	Общее число входов в очередь
ENTRY(0)	Число нулевых входов, без последующего ожидания
AVE.CONT	Среднее количество транзактов в очереди
AVE.TIME	Среднее время пребывания транзактов в очереди
AVE.(-0)	Среднее время пребывания транзактов в очереди без учета нулевых входов
RETRY	Количество транзактов, ожидающих наступления каких-либо условий для входа в очередь

В табл. 10 приведены значения элементов в колонках цепи будущих событий.

Таблица 10. Значения элементов в колонках цепи будущих событий

XN	Номер транзакта
PRI	Приоритет
BDT	Время движения, т. е. время, когда транзакт должен сделать попытку войти в следующий блок
ASSEM	Номер ансамбля
CURRENT	Номер текущего блока
NEXT	Номер следующего блока
PARAMETER	Имена или количество параметров транзакта
VALUE	Значения параметров

Также в отчете могут быть выведены статистические данные по другим объектам GPSS (например, по памятям, таблицам, сохраняемым величинам), содержимое цепи текущих событий и другая информация. Содержимым стандартного отчета можно управлять на вкладке *Reports* (Отчеты) окна *Параметры системы моделирования*, которое вызывается командой **Edit/Settings...** (рис. 27).

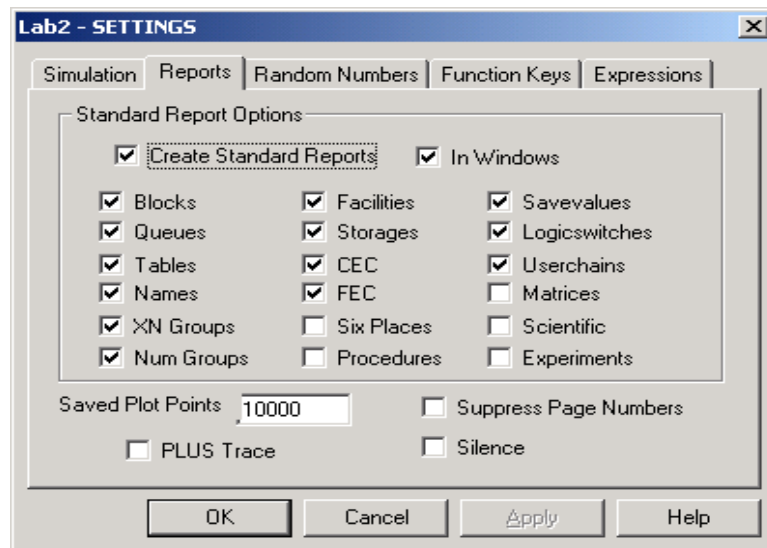


Рис. 27. Задание информации, включаемой в стандартный отчет

3.7. Окно Блоков

Пункт главного меню *Window* служит для управления окнами (рис. 28). В нижней его части показаны все открытые в данный момент окна. В нашем примере это – окно текста модели (*Lab2*), окно *Журнал* (*Lab2.8.sim – JOURNAL*) и окно *Отчет* (*Lab2.8.1 – REPORT*). Активное окно отмечено флажком. После выполнения компиляции модели становится доступной команда *Simulation Window*, которая дает возможность вывести одно из графических окон GPSS для наблюдения за динамикой моделирования (рис. 28).

Познакомимся с окном *Блоков*, задав команду **Window/Simulation Window/Blocks Window**. Его детальный вид показан на рис. 29. В обзорном режиме этого окна будут показаны только значки блоков и транзактов. Переключение режимов с детального на обзорный и наоборот выполняется командой **View/Entity Details** (это верно для всех графических окон).

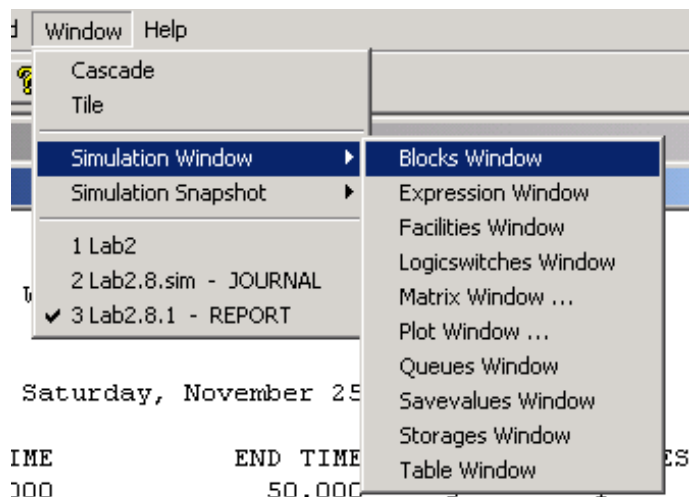


Рис. 28. Пункт меню *Window*

Loc	Block Type	Current Co...	Entry Co...	Retry Ch...	Line Number	Include-file
1 GEN	GENERATE	0	2	0	2	0
2 QUEUE	QUEUE	0	2	0	3	0
3 SEI	SEIZE	0	2	0	4	0
4 DEP	DEPART	0	2	0	5	0
5 ADV	ADVANCE	1	2	0	6	0
6 REL	RELEASE	0	1	0	7	0
7 TER	TERMINATE	0	1	0	8	0
8 GEN	GENERATE	0	1	0	9	0
9 TER	TERMINATE	0	1	0	10	0

Рис. 29. Окно *Блоков*

Транзакт изображается прямоугольником слева от того блока, в котором находится. Если в блоке собирается несколько транзактов, т. е. образуется очередь, то она представляется в виде цветного прямоугольника. При небольшой длине очереди используется зеленый цвет, а при большой – красный. В столбце *Current Count* показывается текущее количество транзактов в блоке, а в столбце *Entry Count* – общее количество транзактов, входивших в данный блок.

В окне *Блоков* можно выполнять простейшие действия по отладке программы модели: прерывать выполнение программы (*Halt*), устанавливать и удалять точки прерывания (*Place* и *Remove*), пошаговое выполнение (*Step*), продолжать имитацию с прерванного места (*Continue*), – см. кнопки на панели инструментов.

► Попытаемся с помощью этих средств восстановить картину событий в нашей модели парикмахерской. Прежде всего, закройте без сохранения файлы отчета и журнала, чтобы начать моделирование сначала. Откомпилируйте модель.

Примечание. Если воспользоваться командой **CLEAR**, то моделирование начинается сначала, но генераторы случайных чисел не приводятся в исходное состояние и результаты дальнейших действий могут отличаться от описанных в данном практикуме.

Активизируйте окно *Блоков*, если оно не является активным. Как обсуждалось в лабораторной работе 2, основные события в системе происходят в момент прихода нового транзакта и в момент окончания обслуживания. Поэтому установите две точки прерывания: на первый блок **GENERATE** и на блок



RELEASE. Для этого нужно выделить мышью соответствующий блок и нажать кнопку **Place** на панели инструментов окна. Эти действия будут отражены в журнале сообщениями «STOP ,1» и «STOP ,6» (рис. 30). Затем начните моделирование командой **Command/START 1**.

Моделирование будет остановлено перед входом транзакта номер 1 (XN: 1) в первый блок GENERATE. Произойдет это в момент времени 22,108 392, как отмечено в журнале (рис. 30).



Чтобы продолжить моделирование, нажмите кнопку **Continue**. Следующее прерывание произойдет в момент прихода нового транзакта либо в момент окончания обслуживания первого транзакта, в зависимости от того, какое событие произойдет ранее. В нашем примере следующее событие происходит в момент времени 34,394 338, когда транзакт номер 1 должен войти в блок RELEASE, т. е. покинуть парикмахера (рис. 30).

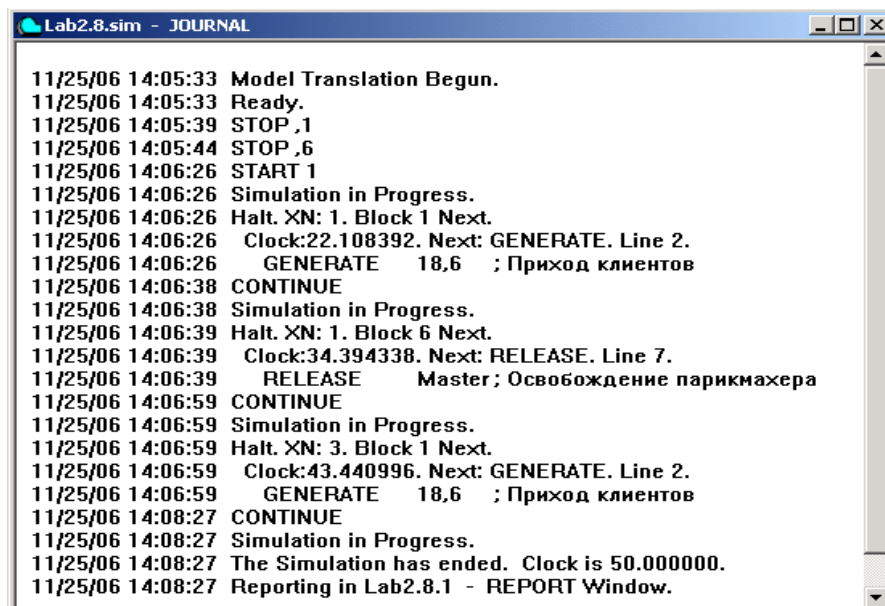


Рис. 30. Содержимое окна Журнал при выполнении прерываний процесса моделирования

Аналогичным способом продолжайте процесс моделирования, нажимая кнопку *Continue* до тех пор, пока не будет закончено моделирование в момент времени 50 и не будет сформирован отчет в окне REPORT. Содержимое этого отчета приведено на рисунках 24–26.

Как следует из данных, приведенных в журнале, в момент времени 43,440 996 в систему приходит третий транзакт. Транзакт номер 2 – это транзакт-таймер, приход которого запланирован в момент начала моделирования на время 50, поэтому следующий транзакт, изображающий клиента парикмахерской, получает номер 3. В момент времени 50 моделирование завершается с приходом транзакта-таймера. Таким образом, на момент окончания моделирования третий транзакт остается в системе в блоке ADVANCE, т. е. обслуживается у парикмахера.

На основании информации журнала можно построить временную диаграмму изменения состояний моделируемой системы (рис. 31).

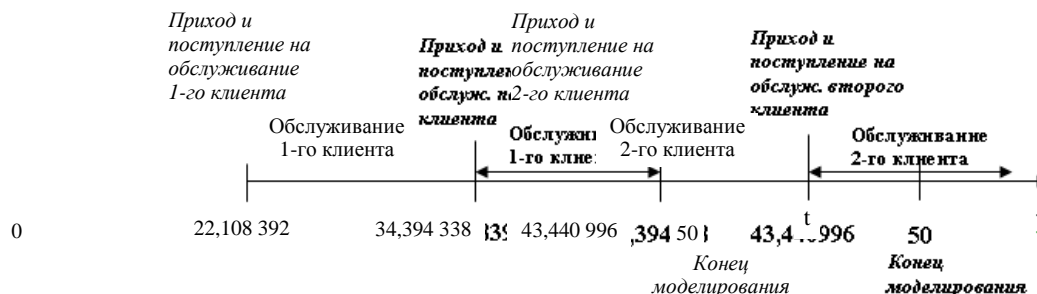


Рис. 31. Временная диаграмма событий в системе

Расчет статистических характеристик очереди QMaster:

- общее число входов в очередь равно 2 – это количество транзактов, прошедших блок QUEUE;
- число нулевых входов также равно 2, так как ни одному транзакту не пришлось ожидать в очереди;
- максимальное количество транзактов в очереди равно 1, так как транзакты регистрировались в очереди;
- среднее значение количества транзактов в очереди равно 0;

- среднее время пребывания в очереди равно 0;
- среднее время пребывания в очереди без учета нулевых входов равно 0 (см. пример расчета этих величин в лабораторной работе 2).

Расчет статистических характеристик устройства Master:

- число обслуживаний равно 2 (учитывается и тот транзакт, который находится на устройстве в момент окончания моделирования);
- коэффициент использования устройства равен $[(34,39 - 22,11) + (50 - 43,44)] / 50 = 0,377$;
- среднее время одного обслуживания равно $[(34,39 - 22,11) + (50 - 43,44)] / 2 = 9,422$.

Очевидно, что эти значения совпадают с результатами моделирования, приведенными в отчете на рис. 26. В отчете также приведена распечатка FEC. Из нее видно, что окончание обслуживания третьего транзакта запланировано на время 53,222 ед., приход в систему четвертого транзакта – на время 58,567 ед., а приход следующего транзакта-таймера – на время 100 ед.

► Теперь следует удалить точки прерывания. Это можно сделать двумя способами:

1. Нажать кнопку *Remove* на панели инструментов окна *Блоки*. Если курсор не стоит на точке прерывания, система задаст вопрос, все ли точки прерывания следует удалить. Нужно ответить утвердительно (*Yes*).

2. Задать команду **Window/Simulation Snapshot/User Stops**. В появившемся диалоговом окне, в котором показаны все точки прерывания с соответствующими номерами блоков, можно выбрать любую точку прерывания и удалить ее кнопкой *Remove*, либо удалить все кнопкой *Remove All*.

3.8. Управляющие операторы GPSS World

Управляющие операторы (команды) могут быть записаны в тексте самой программы, а также могут быть введены через командное меню системы. При этом пункт меню *Command* включает только наиболее употребительные команды, а для ввода остальных управляющих операторов следует задать команду **Command/Custom...** (рис. 32). В появившемся диалоговом окне *Simulation Command* нужно ввести текст команды.

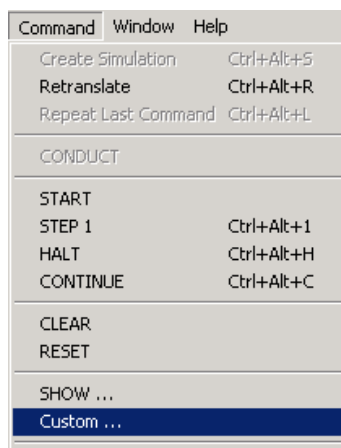


Рис. 32. Меню ввода команд

Список основных управляющих операторов GPSS World приведен в табл. 11.



Кроме того, нажатие некоторых кнопок на панели инструментов в окне блоков фактически равносильно вводу команд. Например, кнопка  равносильна вводу команды STEP 1, а кнопка  соответствует команде HALT и т. д.

Таблица 11. Основные управляющие операторы (команды) языка GPSS

START	Установить счетчик числа завершений и начать моделирование
STEP	Выполнить установленное число попыток входа в блок (пошаговое выполнение)
HALT	Остановить моделирование и удалить все отсроченные команды
CONTINUE	Возобновить прерванное моделирование
CLEAR	Сбросить статистику и удалить транзакты, ранее вошедшие в модель
RESET	Сбросить статистику (транзакты не удаляются)
SHOW...	Вычислить и отобразить значение выражения
EXIT	Закончить сеанс GPSS World
REPORT	Задать имя файлу отчета или выдать отчет немедленно
STOP	Задать условие останова по числу попыток входа в блок
TRACE	Трассировка активного транзакта – каждый вход текущего активного транзакта в новый блок вызовет появление в журнале трассировочного сообщения

Можно также закрепить команды за функциональными клавишами, что обеспечит их быстрое использование. Для этого нужно задать команду **Edit/Settings** и воспользоваться вкладкой *Function Keys* (рис. 33). На этой же вкладке задаются остановки курсора при использовании клавиши *Tab* (Model Tabstops).

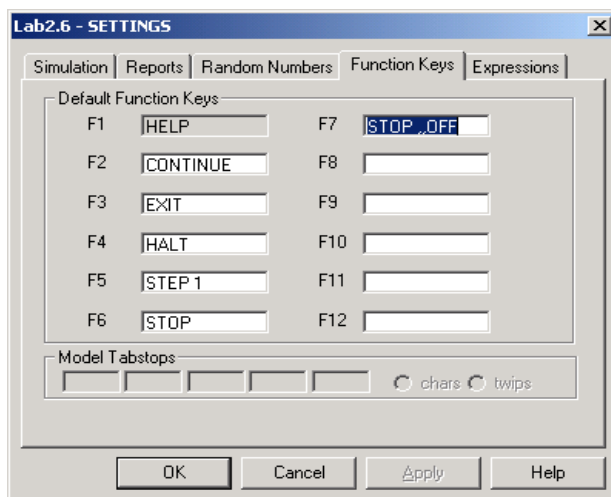


Рис. 33. Окно задания значений функциональных клавиш в модели

3.9. Стандартные числовые атрибуты

Текущее состояние модели GPSS характеризуется набором *стандартных числовых атрибутов* (СЧА). Это некоторые системные переменные, значения которых доступны для пользователя в ходе моделирования. Они могут быть использованы в выражениях и в качестве операндов блоков. СЧА могут относиться как к модели в целом, так и к ее отдельным сущностям. Общими для модели являются:

AC1 – абсолютное системное время (таймер);

C1 – относительное системное время (с последнего RESET);

TG1 – остаток счетчика завершений.

Активный транзакт имеет следующие СЧА:

XN1 – номер;

PR – приоритет;

Pj – значение параметра j;

M1 – резидентное время (таймер минус время прихода транзакта в систему);

MPj – таймер минус значение параметра j.

Подробно СЧА будут обсуждаться при изучении различных блоков GPSS. Те стандартные числовые атрибуты, которые относятся к уже изученным нами блокам, приведены в табл. 12. Их можно обозначить одним из следующих способов, начиная с класса выбранного атрибута:

- по номеру (например, Q2 – текущее количество транзактов в очереди номер 2);
- по имени (например, Q\$QMaster – текущее количество транзактов в очереди с именем QMaster).

Таблица 12. Некоторые стандартные числовые атрибуты

Обозначение класса атрибута	Содержание	Пример
W	Текущее число транзактов в блоке	W2 – текущее число транзактов в блоке 2 (QUEUE)
N	Общее число входов транзактов в блок	N2
F	Состояние устройства: 1 – если занято, 0 – если свободно	F\$Master
FC	Общее количество входов транзактов в устройство	FC\$Master
FR	Коэффициент использования устройства – отношение времени занятости устройства к общему времени моделирования	FR\$Master
FT	Среднее время, в течение которого устройство было занято одним транзактом	FT\$Master
Q	Текущее количество транзактов в очереди	Q\$QMaster
QA	Среднее количество транзактов в очереди	QA\$QMaster
QM	Максимальное количество транзактов в очереди	QM\$QMaster
QT	Среднее время пребывания транзактов в очереди	QT\$Qmaster
QX	Среднее время пребывания транзактов в очереди без учета нулевых входов	QX\$QMaster
QZ	Количество нулевых входов в очередь	QZ\$QMaster

3.10. Просмотр значений СЧА и выражений

Для просмотра значений СЧА и вычисления выражений можно использовать управляющий оператор (команду) SHOW.

► Для примера закройте все открытые окна, кроме окна модели, чтобы начать моделирование сначала. Выполните компиляцию модели, откройте окно *Блоков* и установите точку прерывания на блок RELEASE Master. Начните прогон командой START 1. Имитация будет остановлена, когда первый транзакт попытается покинуть парикмахера, т. е. войти в блок RELEASE. Выполните команду **Command/SHOW** и в диалоговом окне введите SHOW AC1 – обозначение СЧА абсолютного системного времени. После нажатия кнопки OK в строке состояния и в журнале будет выведено 34,394 3 – время окончания обслуживания первого транзакта. Аналогичным путем получите значения следующих СЧА:

- XN1 – номер активного транзакта равен 1;
- M1 – транзитное время активного транзакта равно $12,28 = 34,39 - 22,11$, где 22,11 – время поступления первого транзакта в систему;
- F\$Master – состояние устройства Master равно 1 (занято);
- FR\$Master – коэффициент использования устройства Master равен $0,357 = 12,28 : 34,39$;
- Q\$QMaster – текущее количество транзактов в очереди равно 0.

Продолжите процесс моделирования. После окончания прогона и получения отчета вновь получите значения данных СЧА и прокомментируйте их.

Команду SHOW можно использовать как калькулятор.

► Вызовите *Command/SHOW* и в диалоговом окне введите $1\ 234 + 5\ 678 / 345 \# (64 + 94)$ (роль знака умножения в GPSS играет «#»). Результат 3 834,359 будет высвечен в строке состояния и занесен в журнал.

Другой способ просмотра значений выражений в динамике моделирования – использование окна *Выражений* (Expression Window).

► Выполните команду **Window/Simulation Window/Expression Window**. На экране появится диалоговое окно *Edit Expression Window* (рис. 34), которое предназначено для ввода выражений или СЧА. Например, мы хотим наблюдать изменение системного времени и номер активного транзакта. В поле *Label* напечатайте *Clock* (имя наблюдаемого выражения), а в поле *Expression* – AC1 (значение выражения). Нажмите кнопки *View* и *Memorize*, соответствующие записи появятся в блоках *Window Contents* и *Memorized Expressions*. Первый из них содержит выражения, которые будут видны только в текущем прогоне, а второй – выражения, которые можно будет использовать в будущих прогонах. Исправьте значение в поле *Label* на *Act Trans*, а в поле *Expression* – на XN1, и опять нажмите кнопки *View* и *Memorize*. Для окончания редактирования нажмите кнопку OK. На экране появится окно *Выражений* с двумя записями, как показано на рис. 35.

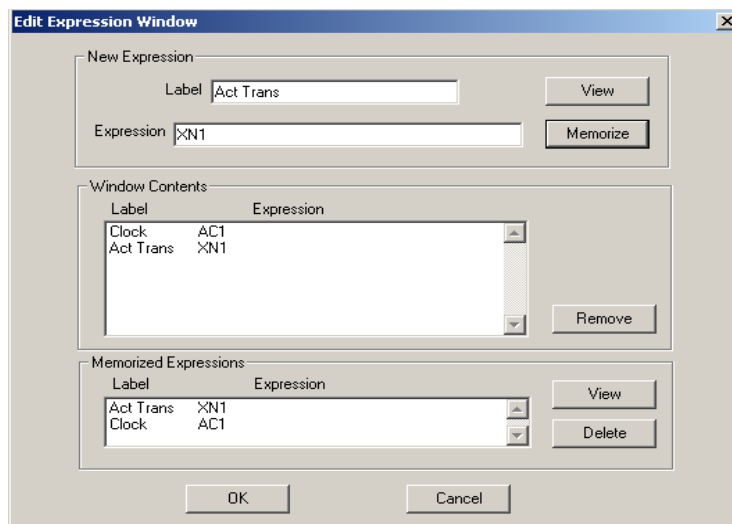


Рис. 34. Окно ввода выражений для просмотра в динамике моделирования

Label	Expression	Value
Clock	AC1	50,000
Act Trans	XN1	3,000

Рис. 35. Окно *Выражений*

Начните моделирование командой **START 1**. Если вы не удалили точку прерывания, поставленную в окне *Блоков*, то для продолжения моделирования можно воспользоваться кнопкой *Continue* на панели инструментов окна *Выражений* или же соответствующей командой в главном меню. Продолжайте моделирование до появления отчета, наблюдая изменение значений выводимых СЧА. Если необходимо отредактировать окно *Выражений* (например, добавить или удалить запись), то можно воспользоваться командой *Edit/Expression Window* при условии, что редактируемое окно должно быть активным.

3.11. Просмотр цепей текущих и будущих событий

Распечатка цепей текущих и будущих событий может быть предусмотрена в отчете (см. рис. 26). Однако пользователь может контролировать состояние этих цепей и в ходе очередного прогона, используя динамические снимки цепей.

► Закройте без сохранения все окна, кроме окна модели. Выполните компиляцию, откройте окно *Блоков* и установите точки прерывания на блоки **GENERATE** (первый по счету) и **RELEASE**, как Вы уже делали при изучении окна *Блоков*. Начните прогон командой **START 1**. Моделирование будет прервано в момент времени 22,108 392, когда первый транзакт попытается войти в блок **GENERATE**. Соответствующая запись появится в журнале. Задайте команду **Window/Simulation Snapshot/CEC Snapshot**. В отдельном окне будет выведено содержимое цепи текущих событий. В ней транзакты сгруппированы по приоритетам. Чтобы раскрыть список каждого приоритета, нужно щелкнуть слева от него по знаку «+». Снимок цепи текущих событий перед приходом в систему первого транзакта показан на рис. 36.



Рис. 36. Снимок цепи текущих событий

Получите также снимок цепи будущих событий командой **Window/Simulation Snapshot/FEC Snapshot** (рис. 37).

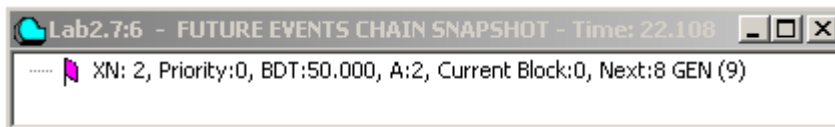


Рис. 37. Снимок цепи будущих событий

Запись информации о каждом транзакте в цепях событий GPSS World несколько отличается от предложенной в лабораторной работе 2. Поясним элементы каждой записи:

- *XN* – номер транзакта;
- *M1* – время поступления транзакта в систему;
- *A* – номер ансамбля (в данной модели он совпадает с номером транзакта);
- *Current Block* – номер текущего блока (если транзакт еще не поступил в систему, он равен нулю);
- *Next* – следующий блок (далее идет сокращенная запись оператора этого блока и в скобках номер строки);
- *Priority* – приоритет;
- *BDT* – время движения транзакта.

Продолжайте процесс имитации и во время каждого останова получите снимки цепей текущих и будущих событий. Каждый снимок относится к определенному моменту времени и не изменяется в ходе имитации. Поэтому старые окна снимков можно закрывать, чтобы не засорять экран. Результаты исследования цепей текущих и будущих событий нашего примера приведены в табл. 13. Поскольку снимки делаются во время фазы коррекции таймера модельного времени, состояние цепей после фазы просмотра списка текущих событий нужно восстановить самостоятельно, аналогично тому, как это делалось в лабораторной работе 2. Сопоставьте данные табл. 13 с временной диаграммой на рис. 31. Прокомментируйте события, которые происходят в каждый момент времени, указанный в таблице. Обратите внимание, что после окончания моделирования в цепи будущих событий остались запланированными события: окончание моделирования третьего транзакта в момент времени 53,222, вход в систему четвертого транзакта-клиента в момент 58,567 и поступление нового транзакта-таймера в момент 100.

Таблица 13. Состояние цепей текущих и будущих событий в процессе моделирования

Время	Цепь текущих событий	Цепь будущих событий
22.108	XN:1, M1:22.108, A:1, Current Block:0, Next:1 GEN(2)	XN:2, Priority:0, BDT:50.000, A:2, Current Block:0, Next:8 GEN(9)
	–	XN:1, Priority:0, BDT:34.394, A:1, Current Block:5, Next:6 REL(7) XN:3, Priority:0, BDT:43.441, A:3, Current Block:0, Next:1 GEN(2) XN:2, Priority:0, BDT:50.000, A:2, Current Block:0, Next:8 GEN(9)
34.394	XN:1, M1:22.108, A:1, Current Block:5, Next:6 REL(7)	XN:3, Priority:0, BDT:43.441, A:3, Current Block:0, Next:1 GEN(2) XN:2, Priority:0, BDT:50.000, A:2, Current Block:0, Next:8 GEN(9)
	–	XN:3, Priority:0, BDT:43.441, A:3, Current Block:0, Next:1 GEN(2) XN:2, Priority:0, BDT:50.000, A:2, Current Block:0, Next:8 GEN(9)
43.441	XN:3, M1:43.441, A:3, Current Block:0, Next:1 GEN(2)	XN:2, Priority:0, BDT:50.000, A:2, Current Block:0, Next:8 GEN(9)
	–	XN:2, Priority:0, BDT:50.000, A:2, Current Block:0, Next:8 GEN(9) XN:3, Priority:0, BDT:53.222, A:3, Current Block:5, Next:6 REL(7) XN:4, Priority:0, BDT:58.567, A:4, Current Block:0, Next:1 GEN(2)
50.000	XN:2, M1:50.000, A:2, Current Block:0, Next:8 GEN(9)	XN:3, Priority:0, BDT:53.222, A:3, Current Block:5, Next:6 REL(7) XN:4, Priority:0, BDT:58.567, A:4, Current Block:0, Next:1 GEN(2)
	–	XN:3, Priority:0, BDT:53.222, A:3, Current Block:5, Next:6 REL(7) XN:4, Priority:0, BDT:58.567, A:4, Current Block:0, Next:1 GEN(2) XN:5, Priority:0, BDT:100.000, A:5, Current Block:0, Next:8 GEN(9)

3.12. Окно Графиков

Модель, которая рассматривалась нами до сих пор, имитирует работу малозагруженного парикмахера. Теоретический коэффициент загрузки системы $\alpha = \frac{10}{18} = 0,56$. В такой ситуации очередь не должна значительно расти. Рассмотрим другой случай. Пусть интервал поступления клиентов в парикмахерскую распределен равномерно 10 ± 3 , а время обслуживания имеет распределение 18 ± 6 . Тогда теоретический коэффициент загрузки $\alpha = \frac{18}{10} = 1,8$ больше единицы. Очередь будет постоянно расти, так как парикмахер не справляется с обслуживанием клиентов. Интервал времени исследования системы за один шаг увеличим до 500.

► Измените соответствующим образом программу модели (рис. 38) и сохраните ее в свою папку под именем *Lab22.gps*.

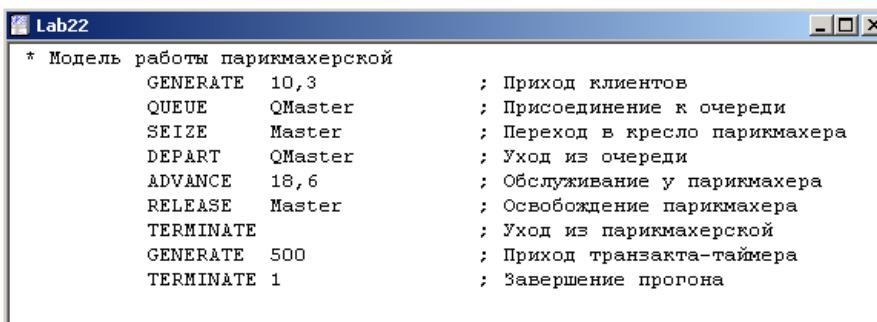


Рис. 38. Модель работы перегруженного парикмахера

Выполните компиляцию модели, откройте окно *Блоков* и начните прогон. Наблюдайте изменение состояния очереди в блоке *QUEUE*. После получения отчета определите максимальную длину очереди, ко-

торая наблюдалась в этом прогоне (в строке QUEUE MAX=21). Эта величина понадобится для оформления графика изменения очереди.

Закройте все окна без сохранения, кроме окна модели. Вновь выполните компиляцию модели и откройте окно *Графиков* командой **Window/Simulation Window/Plot Window**. Диалоговое окно параметров графика заполните, как показано на рис. 39. Поля этого окна имеют следующие значения:

- *Label* – наименование кривой (легенда);
- *Expression* – выводимая зависимость (в нашем случае это СЧА длины очереди QMaster);
- *Title* – наименование графика в целом;
- *Time Range* – максимальное значение системного времени (в нашем случае моделирование прекращается в момент 500 ед. времени, но лучше взять с запасом, например, 600 ед.);
- *Min Value* – минимальное значение по оси Y (длины очереди в нашем случае);
- *Max Value* – максимальное значение по оси Y (как было определено ранее, это значение равно 21, но для красоты графика лучше взять немного больше, например, 30).

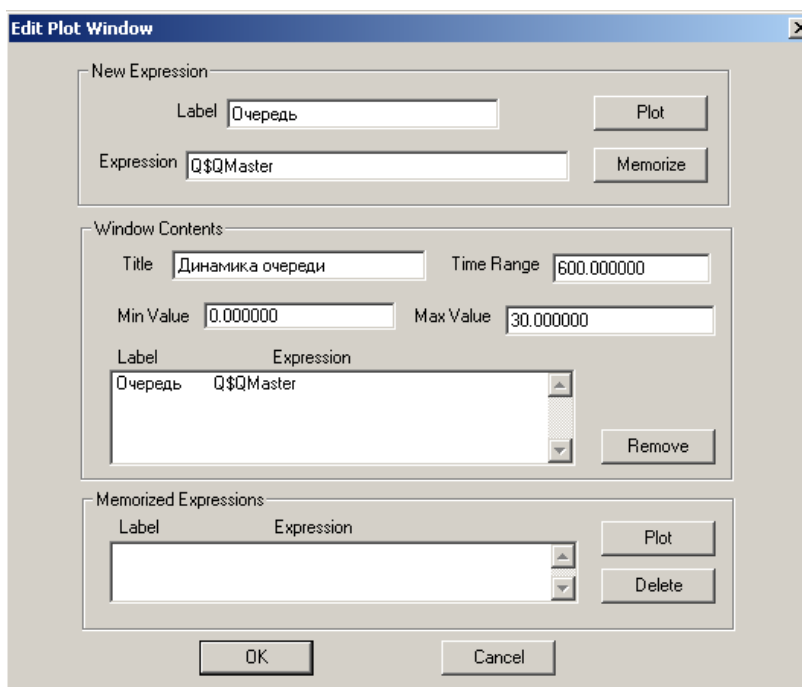


Рис. 39. Диалоговое окно настройки графиков

Закончив ввод параметров графика, нажмите кнопки *Plot* и *OK*. На экране появятся координатные оси. А сам график появится после начала моделирования. Придайте подходящий размер окну *Графиков*, потянув мышью за его угол. Начните прогон модели. После окончания моделирования график будет иметь вид, показанный на рис. 40. Как видно из этого графика, размер очереди постоянно растет.

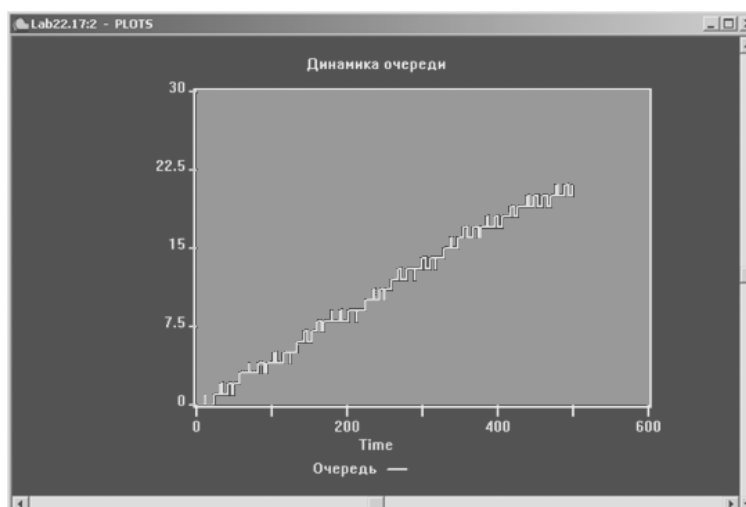


Рис. 40. График изменения очереди

3.13. Окна *Устройств* и *Очередей*

Для просмотра состояния каждого устройства в динамике моделирования используется окно *Устройств* (рис. 41).

► Начните моделирование сначала. Закройте для этого все окна, кроме окна модели, и вновь выполните компиляцию. Откройте окно *Устройств* командой **Window/Simulation Window/Facilities Window** и начните прогон. Состояние устройства Master будет меняться в процессе моделирования, а после его окончания это окно будет содержать следующую информацию (рис. 41), которая появится и в отчете:

- коэффициент использования устройства Master равен 0,976 (*Utilization*);
- в очереди к устройству ожидают 21 транзакт (*Delay Chain*);
- 29 транзактов поступали на обслуживание за прошедшее время (*Acquisitions*);
- устройство доступно (*Available*);
- среднее время обслуживания составляет 16,787 (*Ave. Time*);
- в данный момент на устройстве находится транзакт номер 30 (*Owner XN*);
- в цепях ожидания готовности, ожидания прерывания и прерванных транзактов данного устройства нет ни одного транзакта (*Retry Chain*, *Pending Chain*, *Interrupt Chain*).

Facility	Utilization	Delay Chain	Acquisitions	Available	Ave. Time	Owner XN	Retry Chain
MASTER	0.976	21	29	+	16.787	30	

Рис. 41. Окно *Устройств*

Для просмотра состояния очередей модели используется окно *Очередей*, которое вызывается командой **Window/Simulating Window/Queues Window**.

► Изучите работу этого окна, начав моделирование сначала. Состояние окна *Очередей* по завершению одного прогона показано на рис. 42.

Queue Entity	Current C...	Entry C...	Zero Entry C...	Maximum ...	Average ...	Average Time...	Average Time (-0)
QMASTER	21	50	1	21	10.811	107.871	110.072

Рис. 42. Окно *Очередей*

Состояние очереди QMaster на конец моделирования можно интерпретировать следующим образом:

- текущее количество транзактов в очереди 21 (*Current Count*);
- всего за время моделирования было 50 входов в очередь (*Entry Count*);
- из них один нулевой вход, без ожидания (*Zero Entry Count*);
- максимальное количество транзактов в очереди за время моделирования равно 21 (*Maximum Content*);
- среднее количество транзактов в очереди составляет 10,811 (*Average Content*);
- среднее время ожидания в очереди равно 107,871 (*Average Time*);
- среднее время ожидания без учета нулевых входов 110,072 (*Average Time(-0)*);
- в цепи ожидания готовности нет ни одного транзакта (*Retry Chain*).

Кнопки на панелях инструментов окон *Устройств* и *Очередей* аналогичны кнопкам окна *Блоков* и позволяют остановить процесс моделирования, продолжить его в обычном либо пошаговом режиме. Для окон *Устройств* и *Очередей* можно отключить (включить) детальный режим командой **View/Entity Details**.

3.14. Задания для самостоятельной работы

Рассматривается модель работы парикмахерской с одним мастером. В каждом задании приведены интервалы времени между приходами клиентов (равномерное распределение), интервалы времени обслуживания (равномерное распределение), а также время окончания моделирования.

Введите текст модели в систему GPSS World и сохраните его в своей папке под именем *Задание2.gps*.

Выполните компиляцию и прогон модели. Получите и сохраните стандартный отчет (REPORT) под именем *Отчет Задание2.gpr*, объясните смысл его элементов.

Используя окно *Блоков* и *Журнал*, восстановите последовательность событий в системе и постройте временную диаграмму. Содержимое журнала (JOURNAL) сохраните под именем *Журнал Задание2.sim*.

На основании временной диаграммы рассчитайте характеристики очереди и устройства, сравните их с результатами, приведенными в отчете.

Используя окно *Блоков* и снимки цепей текущих и будущих событий, постройте таблицу состояния цепей в моменты свершения событий в системе (см. раздел 3.11).

Измените программу модели, поменяв местами интервалы времени прихода клиентов и времени обслуживания. Время окончания моделирования увеличьте в 10 раз. Сохраните модель в своей папке под именем *Задание22.gps*.

Выполните компиляцию модели, откройте окно *Блоков* и начните прогон. Наблюдайте изменение состояния системы в окне *Блоков*. Получите и сохраните под именем *Отчет Задание22.gpr* стандартный отчет для этой модели, объясните смысл его элементов.

Постройте график изменения длины очереди для модели *Задание22.gps*.

Наблюдайте работу модели *Задание22.gps*, используя окна *Устройств* и *Очередей*.

Отчет в тетради для лабораторных работ должен содержать:

- формулировку задания для своего варианта;
- распечатку текста модели *Задание2.gps*;
- распечатку отчета *Отчет Задание2.gpr*;
- распечатку журнала (*Журнал Задание2.sim*) и соответствующую временную диаграмму событий в системе;
- расчет характеристик очереди и устройства;
- таблицу состояния цепей текущих и будущих событий в системе;
- распечатку текста модели *Задание22.gps*;
- распечатку отчета *Отчет Задание22.gpr*;
- распечатку графика изменения длины очереди для модели *Задание22.gps*.

Вариант 1

Интервал времени между приходами клиентов 45 ± 10 .

Интервал времени обслуживания клиентов 40 ± 20 .

Предельное значение модельного времени 200.

Вариант 2

Интервал времени между приходами клиентов 20 ± 18 .

Интервал времени обслуживания клиентов 15 ± 8 .

Предельное значение модельного времени 100.

Вариант 3

Интервал времени между приходами клиентов 10 ± 4 .

Интервал времени обслуживания клиентов 6 ± 2 .

Предельное значение модельного времени 50.

Вариант 4

Интервал времени между приходами клиентов 30 ± 8 .

Интервал времени обслуживания клиентов 25 ± 24 .

Предельное значение модельного времени 120.

Вариант 5

Интервал времени между приходами клиентов 5 ± 3 .

Интервал времени обслуживания клиентов 4 ± 2 .

Предельное значение модельного времени 35.

Вариант 6

Интервал времени между приходами клиентов 40 ± 6 .

Интервал времени обслуживания клиентов 30 ± 20 .

Предельное значение модельного времени 200.

Вариант 7

Интервал времени между приходами клиентов 15 ± 5 .

Интервал времени обслуживания клиентов 13 ± 8 .

Предельное значение модельного времени 70.

Вариант 8

Интервал времени между приходами клиентов 50 ± 10 .

Интервал времени обслуживания клиентов 45 ± 20 .

Предельное значение модельного времени 200.

Контрольные вопросы

1. Какие типы файлов используются в системе GPSS World?
2. Как правильно оформить программу модели? Какие есть возможности автоматизации ввода текста модели в системе GPSS World?
3. Компиляция программы модели. Исправление ошибок компиляции.
4. Для чего предназначено окно *Журнал*?
5. Выполнение программы имитационной модели. Понятие прогона. Назначение команды CLEAR.
6. Как должна быть модифицирована программа имитации работы парикмахерской в случае:
 - а) если нужно закончить моделирование по истечении определенного времени (например, 100 ед.)?
 - б) если нужно закончить моделирование, когда через систему пройдет определенное количество транзактов (например, 100)?
7. Общая структура стандартного отчета. Значения элементов в строке общей информации о модели, таблице символических имен и в списке блоков.
8. Что обозначают элементы в таблице информации об устройствах стандартного отчета?
9. Что обозначают элементы в таблице информации об очередях стандартного отчета?
10. Что обозначают элементы в распечатке цепи будущих событий стандартного отчета?
11. Каким образом в GPSS World можно управлять содержимым стандартного отчета?
12. Как может быть вызвано на экран окно *Блоков*? Какая информация представлена в этом окне? В каких режимах может осуществляться просмотр окна *Блоков*?
13. Назначение кнопок на панели инструментов окна *Блоков*.
14. Какими способами могут быть заданы управляющие операторы (команды) GPSS World? Какие основные команды Вы знаете?
15. Что такое стандартные числовые атрибуты? Как можно соотнести СЧА с конкретным объектом GPSS?
16. Какая команда GPSS World позволяет просмотреть значение СЧА или выражения однократно? Как организовать просмотр СЧА или выражений в течение всего времени моделирования?
17. Каким образом в GPSS World можно просмотреть содержимое цепей текущих и будущих событий?
18. Назначение окна *Графиков*. Элементы окна настройки графиков.
19. Назначение и структура окна *Устройств*.
20. Назначение и структура окна *Очередей*.

ЛАБОРАТОРНАЯ РАБОТА 4 ПРИЕМЫ ПРОГРАММИРОВАНИЯ МОДЕЛЕЙ (НАЧАЛО)

4.1. Организация обработки различных типов транзактов

Для моделирования транзактов различных типов, которые обслуживаются на одном устройстве, можно использовать различные сегменты модели. При этом использование одного и того же имени устройства в разных сегментах означает фактически наличие одного устройства, к которому транзакты различных типов выстраиваются в общую очередь. Для организации сбора статистики по этой очереди используется одно и то же имя очереди в блоках QUEUE и DEPART в разных сегментах модели.

Пример 4.1. В парикмахерскую с одним мастером могут приходить клиенты двух типов. Клиентам первого типа нужна только стрижка. Распределение интервалов их прихода (35 ± 10) мин. Клиентам второго типа нужно постричься и побриться. Распределение интервалов их прихода (60 ± 20) мин. Парикмахер обслуживает клиентов в порядке «первым пришел – первым обслужен». Такая ситуация изображена на рис. 43, где кружками обозначены клиенты, желающие только стрижься, а квадратами – желающие постричься и побриться. На стрижку уходит (18 ± 6) мин, а на бритье (10 ± 2) мин. Необходимо написать модель на языке GPSS, обеспечивающую сбор данных об очереди, образуемой ожидающими клиентами.



Рис. 43. Структура моделируемой системы с двумя типами транзактов

На рис. 44 показана блок-схема модели парикмахерской с двумя типами клиентов. Она включает два независимых сегмента, которые моделируют обслуживание клиентов. Первый сегмент соответствует только стригущимся клиентам, а второй – стригущимся и бреющимся. В каждом сегменте пара блоков QUEUE – DEPART описывает одну и ту же очередь, поэтому по транзактам, проходящим в каждом сегменте, обеспечивается совместный сбор статистики. Таким же образом пара блоков SEIZE – RELEASE в каждом из сегментов описывает одно и то же устройство и моделирует самого парикмахера. За единицу времени принята 1 мин.

На рис. 45 приведена соответствующая программа на языке GPSS, а на рис. 46 – результаты одного прогона этой модели, которые показывают, что парикмахер был занят всего 19 раз (количество входов в устройство Master). При этом не делается различий, сколько клиентов относилось к типу только стригущихся, а сколько к другому типу. Очередь к парикмахеру также описывается в обобщенном (агрегированном) виде.

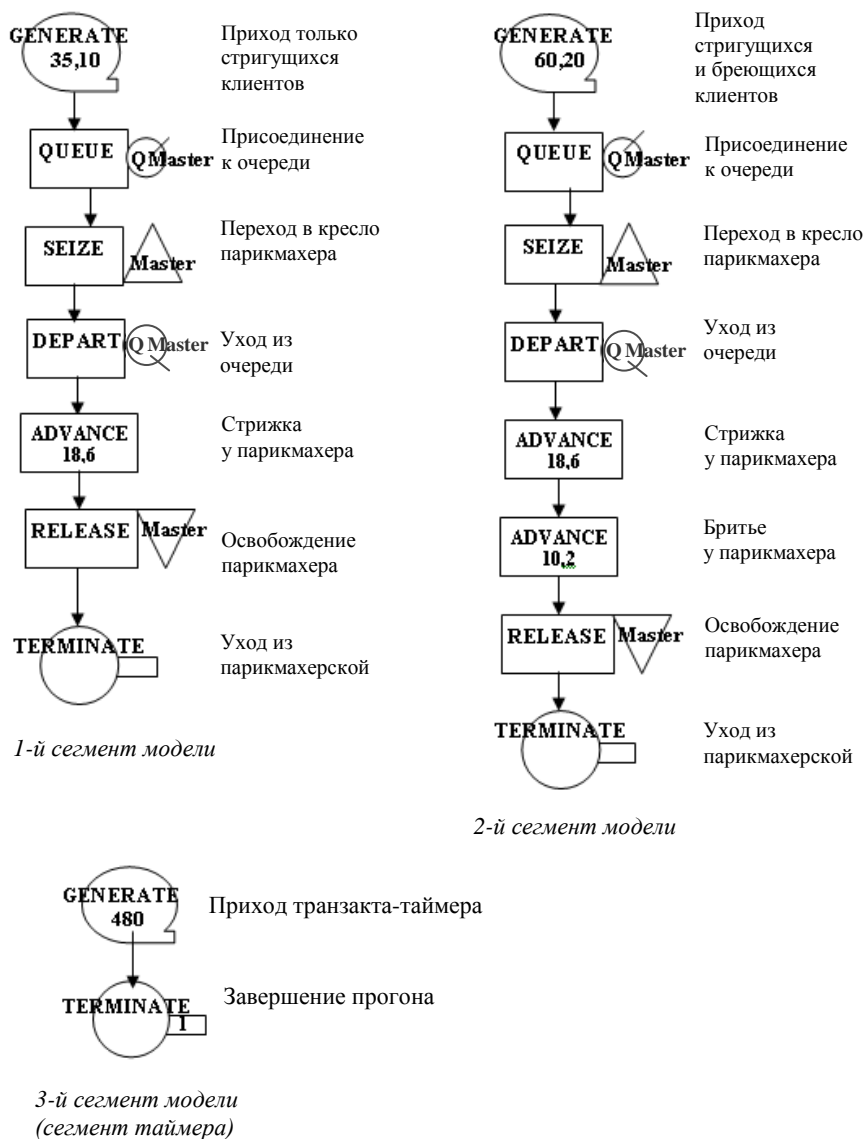


Рис. 44. Блок-схема модели парикмахерской с двумя типами клиентов

```

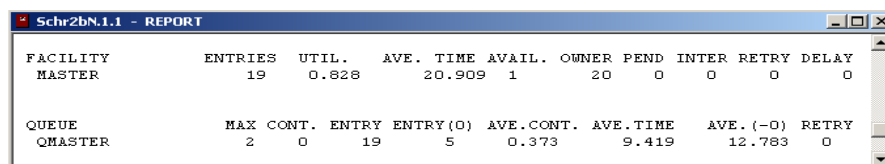
* Модель парикмахерской с клиентами двух типов
*
*      1-й сегмент модели
GENERATE  35,10    ;Приход стригущихся клиентов
QUEUE    QMaster  ;Регистрация в очереди
SEIZE    Master    ;Переход в кресло парикмахера
DEPART   QMaster  ;Уход из очереди
ADVANCE  18,6     ;Стрижка у парикмахера
RELEASE  Master    ;Освобождение парикмахера
TERMINATE                ;Уход из парикмахерской

*
*      2-й сегмент модели
GENERATE  60,20    ;Приход стригущ. и бреющ. клиентов
QUEUE    QMaster  ;Регистрация в очереди
SEIZE    Master    ;Переход в кресло парикмахера
DEPART   QMaster  ;Уход из очереди
ADVANCE  18,6     ;Стрижка у парикмахера
ADVANCE  10,2     ;Бритье у парикмахера
RELEASE  Master    ;Освобождение парикмахера
TERMINATE                ;Уход из парикмахерской

*
*      3-й сегмент модели
GENERATE  480      ;Приход транзакта-таймера
TERMINATE 1        ;Завершение прогона

```

Рис. 45. Программа модели парикмахерской с двумя типами клиентов



FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
MASTER	19	0.828	20.909	1	20	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
QMASTER	2	0	19	5	0.373	9.419	12.783

Рис. 46. Фрагмент отчета для одного прогона модели парикмахерской с клиентами двух типов

Допустим, необходимо собрать статистику ожидания обслуживания как отдельно по каждому типу клиентов, так и по обоим типам вместе. Для этого нужно ввести по одной дополнительной очереди в каждый сегмент (1-й и 2-й). Напомним, что введение объекта *очередь* в модель не означает образование новой очереди ожидающих транзактов, а служит для сбора статистики по уже существующим очередям. Соответствующая программа на языке GPSS приведена на рис. 47.

```

* Модель парикмахерской с клиентами двух типов. Сбор статистики по каждому типу
*
*      1-й сегмент модели
GENERATE  35,10    ;Приход стригущихся клиентов
QUEUE    QMaster  ;Присоед. к очереди общ. стат.
QUEUE    Type1    ;Присоед. к очереди раздел. стат.
SEIZE    Master    ;Переход в кресло парикмахера
DEPART   Type1    ;Уход из очереди раздел. стат.
DEPART   QMaster  ;Уход из очереди общ. стат.
ADVANCE  18,6     ;Стрижка у парикмахера
RELEASE  Master    ;Освобождение парикмахера
TERMINATE                ;Уход из парикмахерской

*
*      2-й сегмент модели
GENERATE  60,20    ;Приход стригущ. и бреющ. клиентов
QUEUE    QMaster  ;Присоед. к очереди общ. стат.
QUEUE    Type2    ;Присоед. к очереди раздел. стат.
SEIZE    Master    ;Переход в кресло парикмахера
DEPART   Type2    ;Уход из очереди раздел. стат.
DEPART   QMaster  ;Уход из очереди общ. стат.
ADVANCE  18,6     ;Стрижка у парикмахера
ADVANCE  10,2     ;Бритье у парикмахера
RELEASE  Master    ;Освобождение парикмахера
TERMINATE                ;Уход из парикмахерской

*
*      3-й сегмент модели
GENERATE  480      ;Приход транзакта-таймера
TERMINATE 1        ;Завершение прогона

```

Рис. 47. Программа модели с дифференцированным сбором статистики по ожиданию клиентов различных типов

Дополнительные очереди Type1 и Type2 введены соответственно в первом и во втором сегментах. Когда транзакт входит в первый сегмент, он сначала регистрируется в очереди QMaster, а потом в очереди Type1. Далее он пытается занять устройство. Если попытка успешна, то транзакт регистрирует уход из очередей Type1 и QMaster, и начинается обслуживание блоком ADVANCE. Таким образом, при ожидании обслуживания только стригущиеся клиенты имеют запись одновременно в двух очередях. То же самое можно сказать о втором типе клиентов, моделирование которых осуществляется во втором сегменте.

Результаты прогона этой модели приведены на рис. 48. Хотя статистика регистрируется по трем очередям, реально существует только одна очередь к обслуживающему устройству – парикмахеру. Анализируя эти результаты, можно сказать, что из 19 клиентов, ожидавших в очереди, 11 были первого типа (только стригущиеся), а 8 – второго типа (стригущиеся и бреющиеся).

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
MASTER	19	0.828	20.909	1	20	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
QMASTER	2	0	19	5	0.373	9.419	12.783	0
TYPE1	1	0	11	4	0.179	7.819	12.288	0
TYPE2	1	0	8	1	0.194	11.619	13.279	0

Рис. 48. Фрагмент отчета для модели дифференцированного сбора статистики по ожиданию клиентов двух типов

Задание 4.1. Выполните следующее:

- из папки *Примеры* загрузите модель *Пример 41.gps*, выполните один прогон модели и изучите отчет;
- модифицируйте модель, чтобы статистика по ожиданию в очереди регистрировалась отдельно по каждому типу клиентов, получите отчет и сравните его с отчетом на рис. 48;
- доработайте модель так, чтобы она измеряла процент времени, который парикмахер затрачивает на обслуживание клиентов каждого типа, и среднее время обслуживания клиентов отдельно по каждому типу;
- программу и отчет по последней модели приведите в тетради для лабораторных работ.

4.2. Использование приоритетов транзактов

В системе GPSS World по умолчанию принята дисциплина обслуживания очереди «первым пришел – первым обслужен внутри приоритетного класса». Это означает, что в очереди транзакты выстраиваются согласно приоритету. В начале очереди располагаются транзакты, обладающие наивысшим приоритетом, а в конце – низкоприоритетные транзакты. В первую очередь на обслуживание выбираются транзакты из более приоритетного участка очереди, и только в случае их отсутствия – из менее приоритетного участка. Транзакты с одинаковым приоритетом обслуживаются по принципу «первым пришел – первым обслужен». Данная дисциплина очереди схематично изображена на рис. 49.



Рис. 49. Дисциплина обслуживания «первым пришел – первым обслужен внутри приоритетного класса»

Приоритет присваивается транзакту в момент его создания блоком GENERATE. Параметр E этого блока задает числовое значение приоритета создаваемого транзакта. Например, если транзакты должны поступать в модель с интервалом (420 ± 360) ед. модельного времени и им должен присваиваться приоритет 1, то блок создания транзактов имеет следующий вид (параметры C и D принимают значения по умолчанию):

GENERATE 420,360,,1

Далее в модели приоритет активного транзакта может быть изменен с помощью блока

PRIORITY A

Операнд A – значение присваиваемого транзакту приоритета, от 0 до 127 включительно.

Приоритету активного транзакта соответствует стандартный числовой атрибут PR, просмотреть значение которого можно командой

SHOW PR

Пример 4.2. На некоторой фабрике на складе работает один кладовщик. Он выдает запасные части механикам, обслуживающим станки и устанавливающим эти части на станках, подлежащих ремонту. Время, необходимое для удовлетворения запроса, зависит от типа запасной части. Запросы бывают двух категорий. Соответствующие данные приведены в табл. 14.

Таблица 14. Характеристики запросов механиков на обслуживание

Категории запроса	Интервалы времени прихода механиков, с	Время обслуживания, с
1-я	420 ± 360	300 ± 90
2-я	360 ± 240	100 ± 30

Кладовщик обслуживает механиков по принципу «первым пришел – первым обслужен» независимо от категории запроса. Поскольку сломанный станок ничего не производит, то простой механика в очереди приносит убыток 0,25 цента в секунду (9 долл. США в час). Эта стоимость не зависит от того, за какой запасной частью пришел механик. Руководитель считает, что среднее число простаивающих механиков можно уменьшить, если запросы второй категории в кладовой будут удовлетворяться быстрее запросов первой категории (по принципу «ближайшая операция – кратчайшая»).

Необходимо создать модель работы склада для обеих дисциплин обслуживания очереди, выполнить моделирование для каждой из них в течение 8-часового рабочего дня и узнать, уменьшится ли среднее число ожидающих механиков в очереди, и сколько денег ежедневно будет сэкономлено при использовании приоритетного обслуживания. В стоимость потерь не следует включать простои механиков во время обслуживания.

Модель этого примера для случая, когда не делается различия между категориями запросов, полностью аналогична примеру 4.1. Если же один из запросов имеет преимущество в очередности обслуживания, это можно реализовать, назначив соответствующему транзакту более высокий приоритет. Например, запросам первой категории назначим приоритет 1, а запросам второй категории – 2 (числовые значения приоритетов не играют роли, главное, чтобы второй был больше первого). Необходимый порядок выбора транзактов из очереди на обслуживание (транзакты высшего приоритета имеют преимущество) реализуется в GPSS автоматически. Программа модели представлена на рис. 50. Единицей времени является 1 с. Время моделирования соответствует 8-часовому рабочему дню: $8 \cdot 60 \cdot 60 = 28\,800$ с.

```

*      Модель работы склада
*
*      Механики, делающие запрос категории 1
GENERATE 420,360,,,1 ;Прибытие на склад
QUEUE   Line         ;Присоед. к очереди
SEIZE    Clerk        ;Занятие кладовщика
DEPART   Line         ;Уход из очереди
ADVANCE  300,90       ;Обслуж. кладовщиком
RELEASE  Clerk        ;Освобожд. кладовщика
TERMINATE                                ;Уход со склада
*
*      Механики, делающие запрос категории 2
GENERATE 360,240,,,2 ;Прибытие на склад
QUEUE   Line         ;Присоед. к очереди
SEIZE    Clerk        ;Занятие кладовщика
DEPART   Line         ;Уход из очереди
ADVANCE  100,30       ;Обслуж. кладовщиком
RELEASE  Clerk        ;Освобожд. кладовщика
TERMINATE                                ;Уход со склада
*
*      Таймер
GENERATE 28800        ;Таймер через 8 ч
TERMINATE 1           ;Прекращ. моделирования

```

Рис. 50. Модель работы склада с приоритетами запросов

Сравнение результатов моделирования для двух способов организации очереди приведено в табл. 15. Простой каждого механика в очереди обходится в 0,25 цента в секунду, или $0,25 \cdot 60 \cdot 60 = 900$ центов, рав-

ных 9 долл. США в час, или $9 \cdot 8 = 72$ долл. США в день. Поэтому, чтобы получить общие дневные потери, нужно умножить количество простаивающих рабочих на 72. Таким образом, при использовании приоритетного обслуживания будет сэкономлено $334 - 188 = 146$ долл. США.

Таблица 15. Сравнительный анализ дневных потерь при различных способах организации очереди

Способы организации очереди	Среднее число механиков в очереди	Дневные потери от простоя механиков, долл. США
Без приоритетов	4,637	$4,637 \cdot 72 \approx 334$
С приоритетами	2,616	$2,616 \cdot 72 \approx 188$

Пример 4.3. В модели работы парикмахерской, которая рассматривалась в лабораторной работе 2, парикмахер работает 480 мин без перерыва. А теперь рассмотрим модель парикмахерской, которая открывается в 8 ч 30 мин, а закрывается в 17 ч. Парикмахер имеет перерыв на обед в 12 ч или после этого времени: если он простаивал в полдень, то получает перерыв ровно в 12 ч, а если он был занят обслуживанием клиента, то получает перерыв сразу же после окончания обслуживания. Клиенты, которые приходят в парикмахерскую в течение перерыва, ждут его возвращения. Укажите, какой будет при этом нагрузка парикмахера.

Для решения этой задачи в модель следует ввести отдельный сегмент, имитирующий звонок на перерыв. Транзакт-перерыв поступает в модель ровно в полдень и имеет приоритет выше, чем приоритет транзактов, изображающих клиентов (например, равный одному, а клиенты по умолчанию имеют нулевой приоритет). Такой транзакт займет устройство сразу же после его освобождения. Время обслуживания транзакта-перерыва детерминированное и равно 30 мин. Общее время моделирования нужно изменить с 480 на 510 (с 8 ч 30 мин до 17 ч проходит 510 мин). Текст модели показан на рис. 51. К сожалению, такой способ моделирования обеденного перерыва имеет побочный эффект: в статистику по устройству Master будет включено время обеденного перерыва, как будто оно было занято обслуживанием клиентов. Подумайте, как нужно модифицировать модель, чтобы получить статистику обслуживания устройством только тех транзактов, которые имитируют клиентов.

```

*      Модель работы парикмахерской
*
* Сегмент моделирования клиентов
GENERATE      18,6      ;Приход клиентов
QUEUE         QMaster   ;Присоед. к очереди
SEIZE         Master    ;Переход в кресло парикм.
DEPART        QMaster   ;Уход из очереди
ADVANCE       16,4      ;Обслуж. у парикмахера
RELEASE       Master    ;Освобожд. парикмахера
TERMINATE     ;Уход из парикмахерской
*
* Сегмент таймера перерыва
GENERATE      210,,1,1  ;Звонок на перерыв
SEIZE         Master    ;Занятие парикмахера
ADVANCE       30        ;Обед полчаса
RELEASE       Master    ;Освобожд. парикмахера
TERMINATE     ;Уничтож. транзакта-таймера
*
* Сегмент таймера времени моделирования
GENERATE      510       ;Приход таймера окончания
TERMINATE     1         ;Завершение прогона

```

Рис. 51. Моделирование обеденного перерыва парикмахера

Задание 4.2. Из папки *Примеры* загрузите модель работы парикмахерской *Пример43.gps*. Добавьте еще два 15-минутных перерыва на кофе: в 10 ч и в 15 ч (если в этот момент обслуживается клиент, то его обслуживание должно быть закончено, прежде чем мастер приступит к кофе). Работа должна начинаться в 8 ч и заканчиваться в 17 ч. Клиенты, приходящие в парикмахерскую во время перерывов или ожидавшие ранее, продолжают ожидать освобождения мастера. Определите, какой будет нагрузка парикмахера, учитывая, что время перерывов не должно считаться временем обслуживания.

4.3. Изменение маршрутов движения транзактов

Для изменения маршрута транзактов в модели применяются блоки TRANSFER, GATE, TEST, LOOP. Транзакты, входящие в эти блоки, продвигаются не к следующему блоку, а к блокам, адрес которых определяется либо указывается в операторах. В данном практикуме рассматривается только блок TRANSFER.

Блок TRANSFER (*Переместить*) может иметь 4 операнда (A,B,C,D). Существуют девять режимов использования этого блока, которые определяются по значению операнда A. Рассмотрим основные из них:

- *Безусловный переход:*

TRANSFER ,B

Здесь B – имя (адрес) блока, к которому направляется транзакт, операнд A отсутствует.

Примеры

TRANSFER ,CPU2

TRANSFER ,*+3

TRANSFER ,CPU1+2

Первый оператор направляет транзакт к блоку с именем CPU2. Два остальных оператора – примеры относительной адресации. Второй оператор посылает транзакт к третьему по счету блоку после текущего, а последний оператор направляет транзакт ко второму по счету блоку от блока CPU1.

- *Условный переход с одним альтернативным адресом:*

TRANSFER BOTH,B,C

Операнды имеют следующие значения:

BOTH – тип режима;

B – имя блока, к которому направляется транзакт (основной адрес);

C – альтернативный адрес, куда направляется транзакт при отсутствии возможности войти в блок с адресом B.

Если параметр B опущен, то по умолчанию транзакт направляется к следующему по порядку блоку.

Если ни один из блоков, указанных в операторе TRANSFER, не может принять транзакт, то он остается в блоке TRANSFER. При каждом последующем просмотре цепи текущих событий система пытается ввести транзакт в блок B, а если его вход по-прежнему невозможен, то в блок C.

Примеры

TRANSFER BOTH,KASS1,KASS2

TRANSFER BOTH,TWO

В первом операторе транзакт первоначально направляется к блоку с именем (меткой) KASS1. При невозможности войти в этот блок транзакт направляется к блоку с именем KASS2. Во втором операторе транзакт делает попытку войти в следующий блок (значение параметра B по умолчанию), а если ему это не удастся, он направляется к блоку с именем TWO.

- *Условный переход с многими альтернативами:*

TRANSFER ALL,B,C,D

Операнды имеют следующие значения:

ALL – тип режима;

B – первый адрес;

C – последний адрес;

D – константа M, используемая для вычисления возможных адресов движения транзактов: адрес в поле B, затем B+M, B+2M, ..., адрес в поле C. Если этот операнд опущен, то по умолчанию он равен единице.

Пример

TRANSFER ALL,FIRST,LAST,3

Транзакт последовательно пытается войти в следующие блоки: FIRST, FIRST+3, FIRST+6, ..., LAST.

- *Статистический переход (переход с заданной вероятностью):*

TRANSFER A,B,C

Операнды имеют следующие значения:

A – вероятность перехода транзакта по адресу C;

B – альтернативный адрес.

Если операнд A представляет собой целое число, то оно трактуется как количество тысячных долей вероятности и преобразуется в вероятность автоматически. При пропущенном параметре B имеется в виду следующий по порядку блок.

Выбор адреса назначения транзакта, розыгрыш, согласно заданной вероятности, происходит один раз, когда транзакт входит в блок TRANSFER. Если выбранный блок не может принять транзакт, то он остается в блоке TRANSFER, пока этот переход не станет возможным.

TRANSFER .370,THIS,THAT

С вероятностью 0,37 транзакт перейдет к блоку THAT, а с вероятностью 0,63 – к блоку THIS.

TRANSFER 750,NEW_PLACE

С вероятностью 0,75 транзакт передается на блок NEW_PLACE, а с вероятностью 0,25 – на следующий блок.

Описание остальных режимов (PICK, FN, P, SBR, SIM) можно изучить в справочнике по GPSS (меню *Help*). Во всех режимах операнд С должен быть больше операнда В, а если используется операнд D, то разность $C - B$ должна делиться на D. Если транзакт не принимает ни один из блоков, куда делалась попытка его «протолкнуть», то он остается в блоке TRANSFER. В этом случае моделирующая программа повторяет попытки переправить транзакт при каждом изменении модельного времени, что может привести к большим затратам времени моделирования. На блок-схеме блок TRANSFER изображается в виде ромба (рисунки 52 и 53).

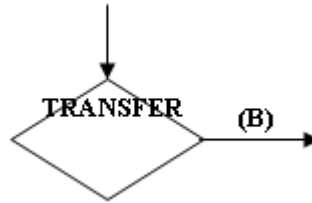


Рис. 52. Блок TRANSFER в режиме безусловной передачи

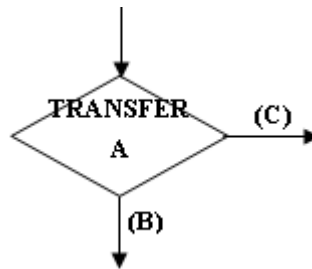


Рис. 53. Блок TRANSFER в режиме статистического перехода или перехода с одним альтернативным адресом

Пример 4.4. Производство деталей определенного вида включает длительный процесс сборки, заканчивающийся коротким периодом обжига в печи. Поскольку содержание печи обходится довольно дорого, несколько сборщиков используют одну печь, которая позволяет обжигать одновременно только одну деталь. Сборщик не может начать новую сборку, пока не вытащит из печи предыдущую деталь. Таким образом, сборщик работает в следующем режиме:

1. Собирает следующую деталь.
2. Ожидает возможности использования печи по принципу «первым пришел – первым обслужен».
3. Использует печь.
4. Возвращается к пункту 1 (собирает следующую деталь).

Стоимость операций и изделий, а также время, необходимое на различные операции, представлены в таблицах 16 и 17.

Необходимо построить на GPSS модель описанного процесса и определить оптимальное число сборщиков, использующих одну печь. Под оптимальным понимают такое число, которое дает максимальную прибыль. Моделирование следует выполнять в течение 40 ч модельного времени. Предполагается, что в течение рабочего дня нет перерывов, а рабочие дни идут подряд без выходных.

Таблица 16. Время на различные операции по изготовлению детали

Операции	Необходимое время, мин
Сборка	30 ± 5
Обжиг	8 ± 2

Таблица 17. Информация о стоимости операций и изделий

Показатели	Стоимость, долл. США
Зарплата сборщика в час	3,75
Стоимость печи за 8-часовой рабочий день	80
Цена материала за одну деталь	2
Стоимость готового изделия за одну деталь	7

При моделировании этой системы печь будет представлять собой устройство с именем Oven, а сборщики – транзакты. Аналогично тому, как они периодически осуществляют сборку и обжиг, транзакты перемещаются в модели системы. Чтобы ограничить общее число транзактов, циркулирующих в модели, нужно использовать операнд D блока GENERATE. Для определения прибыли, соответствующей заданному числу сборщиков, необходимо знать, сколько готовых деталей они сделали в течение моделируемого периода. Это число деталей совпадает с числом раз использования печи, или устройства Oven. За единицу времени примем 1 мин. Тогда время моделирования будет равно $40 \cdot 60 = 2\,400$ мин. Блок-схема модели производства деталей показана на рис. 54. Соответствующая программа на GPSS приведена на рис. 55.

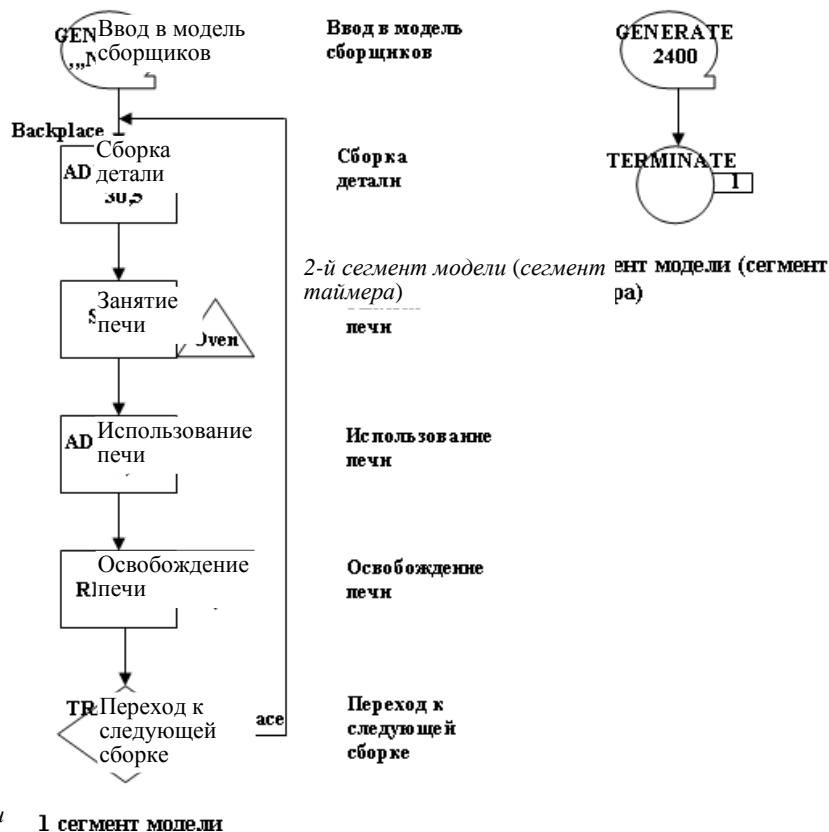


Рис. 54. Блок-схема модели производства деталей

* Модель производства деталей с обжигом в печи
 Number EQU 4 ;Количество сборщиков

* Сегмент моделирования сборщиков
 GENERATE , , , Number ;Ввод в модель сборщиков

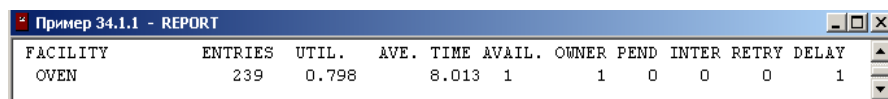
Backplace ADVANCE 30,5 ;Сборка детали
 SEIZE Oven ;Занятие печи
 ADVANCE 8,2 ;Использование печи
 RELEASE Oven ;Освобождение печи
 TRANSFER ,Backplace ;Переход к след. сборке

* Сегмент таймера окончания моделирования
 CENERATE 2400 ;Таймер через 40 ч
 TERMINATE 1 ;Прекращ. моделирования

Рис. 55. Программа модели производства деталей

В этой модели блок ADVANCE 30,5 получил символическое имя Backplace. Оно расположено в поле метки этого оператора и используется в операторе TRANSFER для указания, куда потом нужно отправить транзакт. В модели не использованы блоки QUEUE и DEPART, поскольку не требуется сбор статистики по очереди к печи (очередь при этом все равно существует). Транзакты, ожидающие освобождения устройства, остаются в блоке с именем Backplace, даже если время их задержки уже истекло.

В отчете по этой модели требуется определить число деталей, которые вышли из устройства Oven (количество произведенных деталей). Оно равно количеству входов в устройство Oven (поле ENTRIES) минус 1, если для одной детали еще не закончен обжиг (поле OWNER не равняется нулю). Например, из отчета для случая четырех сборщиков (рис. 56) очевидно, что число произведенных деталей равно $239 - 1 = 238$ шт. Рассчитаем прибыль организации для этого количества деталей. За проданные детали будет получен доход $238(7 - 2) = 1\,190$ долл. США. Стоимость использования печи $80 \cdot 5 = 400$ долл. США за 5 дней. Зарплата сборщиков $4 \cdot 3,75 \cdot 40 = 600$ долл. США за 5 дней. Поэтому чистая прибыль организации $1\,190 - 400 - 600 = 190$ долл. США.



FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
OVEN	239	0.798	8.013	1	1	0	0	0	1

Рис. 56. Отчет по устройству Oven для случая четырех сборщиков

Чтобы определить прибыль для другого количества сборщиков, нужно поменять значение переменной Number в модели, выполнить один прогон и рассчитать прибыль аналогичным способом.

Задание 4.3. Из папки *Примеры* загрузите модель *Пример 44.gps*. Выполните исследование модели для случая четырех, пяти и шести сборщиков. Рассчитайте прибыль организации для каждого случая. Результаты сведите в таблицу и оформите ее в тетради для лабораторных работ. Укажите, какое количество сборщиков является оптимальным, и можно ли считать результаты моделирования достаточно точными. В тетради для лабораторных работ приведите текст программы для одного любого случая, указав, какие изменения в программе следует произвести для других случаев; таблицу результатов расчета прибыли, а также выводы по исследуемой системе.

4.4. Моделирование многоканальных устройств

Многоканальному устройству в системе GPSS соответствует объект *память* (символическое обозначение S). Память может выполнять параллельное обслуживание нескольких транзактов. При этом необходимо, чтобы такое обслуживание было однородным, т. е. обладало некоторыми общими свойствами. Например, среднее время обслуживания и разброс этого времени для кассиров банка, одновременно обслуживающих клиентов, должно быть одинаковым. Если же они обслуживают с разной скоростью, то для их имитации следует использовать несколько одновременно работающих устройств.

Емкость памяти, т. е. максимальное количество одновременно обслуживаемых транзактов (число каналов обслуживания) задается оператором описания STORAGE, который имеет формат

Name STORAGE A

Операнды данного оператора имеют следующие значения:

Name – имя (номер) памяти;

A – емкость памяти.

Пример

1 STORAGE 100 ; память с номером 1 емкости 100
PAM STORAGE 10 ; память с именем PAM емкости 10

Операторы описания памяти обычно помещаются в начале программы перед первым оператором GENERATE.

Изменение состояния памяти в модели осуществляется с помощью операторов занятия и освобождения памяти ENTER и LEAVE:

ENTER A,B
LEAVE A,B

Значения операндов следующие:

A – имя (номер) памяти;

B – число единиц памяти, занимаемых или освобождаемых транзактом при входе в блоки (по умолчанию 1).

Примером, когда один транзакт может потребовать несколько единиц памяти, является моделирование причала в порту. В зависимости от размера корабль может потребовать использование нескольких причалов. Транзакт не может войти в блок ENTER, если нет достаточного количества свободных единиц памяти. Не обязательно освобождается такое же число единиц памяти, какое было занято. Память может освобождаться транзактом, ее ранее не занимавшим, в отличие от устройства. Однако в сумме число освобождаемых единиц не должно превосходить числа занимаемых транзактами единиц памяти.

Блоки ENTER и LEAVE изображаются, как показано на рис. 57.

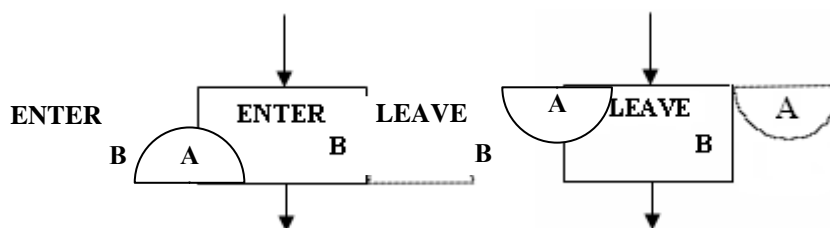


Рис. 57. Блоки ENTER (Войти) и LEAVE (Выйти)

Память моделирует многоканальное устройство, к которому транзакты выстраиваются в общую очередь, обслуживаемую по принципу «первым пришел – первым обслужен внутри приоритетного класса» (рис. 58). Примером является «быстрая очередь» в банке, когда клиент не выбирает себе окошко, а становится в конец общей очереди. Если же перед параллельно обслуживающими каналами возникает несколько очередей, то моделирование их сложнее и будет рассмотрено позже.

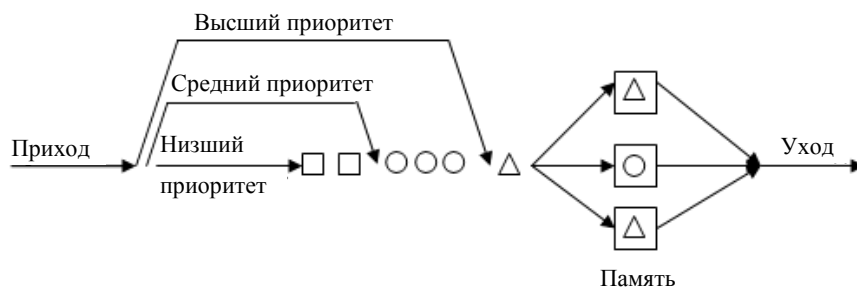


Рис. 58. Многоканальная система массового обслуживания с одной очередью и приоритетами

Связанные с памятью стандартные числовые атрибуты показаны в табл. 18.

Таблица 18. Стандартные числовые атрибуты, связанные с памятью

Обозначение	Содержание	Примеры
R	Количество доступных каналов в памяти j	R\$PAM, Rj
S	Текущее содержимое памяти j, т. е. количество занятых каналов	S\$PAM, Sj
SA	Среднее содержимое памяти j за время моделирования	SA\$PAM, SAj
SC	Общее количество транзактов, входивших в память j	SC\$PAM, SCj
SE	Состояние памяти по незанятости: 1, если память j полностью свободна, и 0 в противном случае	SE\$PAM, SEj
SF	Состояние памяти по занятости: 1, если занят хоть один канал памяти, и 0 в противном случае	SF\$PAM, SFj
SR	Коэффициент использования памяти, вычисляемый как отношение среднего содержимого памяти к ее полному объему и выражаемый в частях от 0 до 1 000	SR\$PAM, SRj
SM	Максимальное количество транзактов, находившихся в памяти j за время моделирования	SM\$PAM, SMj
ST	Среднее время пребывания в памяти одного транзакта	ST\$PAM, STj
SV	Состояние памяти по готовности: 1, если память j готова, и 0, если не готова	SV\$PAM, SVj

В конце моделирования по каждой памяти распечатывается статистика:

- количество свободных каналов памяти на момент окончания моделирования (REM.);
- минимальное и максимальное количество одновременно занятых каналов (MIN. и MAX.);
- счетчик входов (ENTRIES);
- состояние доступности памяти (AVL. равно 1, если устройство доступно, и равно 0, если недоступно);
- среднее число занятых каналов (AVE.C.);
- коэффициент использования памяти (UTIL.) (средняя доля занятых каналов);
- количество транзактов в цепи *Retry Chain* (RETRY);
- количество транзактов в цепи *Delay Chain* (DELAY) – транзакты, ожидающие в очереди к памяти.

Пример 4.5. На трикотажной фабрике 50 швейных машин работают по 8 ч в день и по 5 дней в неделю. Любая из этих машин может в любой момент выйти из строя. В этом случае ее заменяют резервной машиной, сразу или по мере ее появления. Тем временем сломанную машину отправляют в ремонтную мастерскую, где ее чинят и возвращают в цех, но уже в качестве резервной. В существующем замкнутом цикле движения машин можно выделить четыре фазы (рис. 59). Неперечеркнутые квадраты на рисунке представляют собой работоспособные машины, а перечеркнутые – ремонтируемые либо ожидающие ремонта.

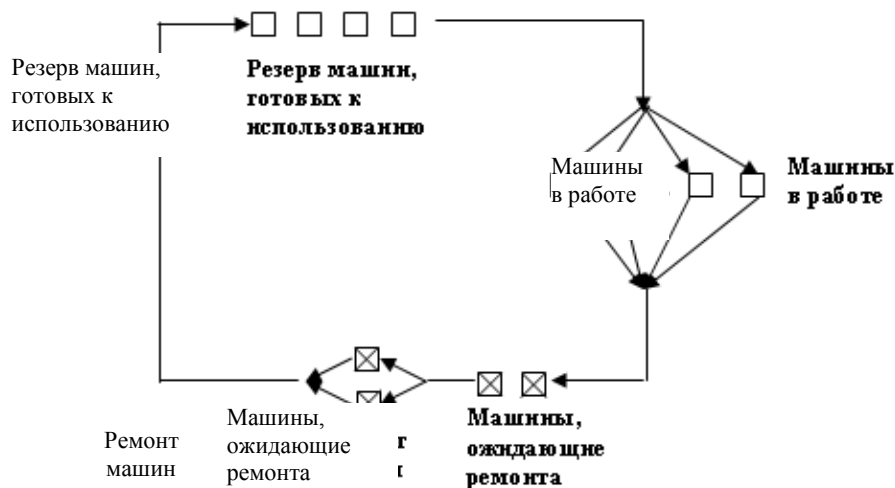


Рис. 59. Функциональная схема примера 4.5

Управляющий хочет знать, сколько рабочих следует нанять для работы в мастерской и сколько машин следует иметь в резерве (резервные машины арендуются), чтобы ими можно было подменить 50 собственных. Цель – минимизация стоимости производства. Оплата рабочих в мастерской 3,75 долл. США в час, за арендуемую машину надо платить по 30 долл. США в день. Почасовой убыток при использовании менее 50 машин в производстве оценивается примерно в 20 долл. США за машину. Этот убыток возникает из-за снижения производства.

Опыт эксплуатации показывает, что на ремонт сломанной машины уходит примерно (7 ± 3) ч, распределение равномерное. Когда машину используют в производстве, время наработки до отказа распределено равномерно и составляет (157 ± 25) ч. Время, необходимое для перевозки машины из цеха в мастерскую и обратно, настолько мало, что его не учитывают.

Плата за арендуемые машины не зависит от того, используют их или они простаивают. По этой причине не делают попыток увеличить число собственных машин в работе. Различия в эксплуатации собственных и арендуемых машин также нет.

Необходимо построить на GPSS модель такой системы и исследовать на ней расходы при различном числе арендуемых машин и рабочих в мастерской в течение трех лет. Следует выбрать наилучший вариант организации системы по критерию минимизации расходов.

Рабочие в мастерской и машины, одновременно находящиеся в производстве, могут быть представлены многоканальными устройствами обслуживания. Поэтому для их моделирования будем использовать памяти с именами Men и Nowon соответственно. Емкость памяти Men равна количеству рабочих в мастерской, которое будет меняться для разных экспериментов на модели (рассмотрим 3, 4 и 5 рабочих). Емкость памяти Nowon постоянна и равна 50. Машины, циркулирующие в системе, моделируются транзактами. Машины, находящиеся в резерве, – это транзакты в очереди к памяти Nowon; а машины, ожидающие ремонта, – это транзакты в очереди к памяти Men.

Обслуживание транзакта одним из каналов памяти Nowon моделируется задержкой на время наработки до отказа (157 ± 25) . Когда машина выходит из строя, она покидает память Nowon и становится в очередь к памяти Men (рабочие мастерской). Ремонт в мастерской имитируется задержкой на (7 ± 3) ч, а затем машина покидает память Men и направляется в резерв (становится в очередь к памяти Nowon). Общее число транзактов, циркулирующих в системе, равно 50 собственных машин плюс некоторое число арендуемых для резерва (рассмотрим случаи аренды трех, четырех и пяти машин). Это число следует задавать до начала прогона модели как ограничитель количества транзактов в блоке GENERATE (переменная Machines). Для ограничения времени моделирования используется сегмент таймера, который приходит в момент 6 240 (8 ч · 5 дней в неделю · 52 недели в году · 3 года). За единицу времени принят 1 ч. Программа модели показана на рис. 60, а соответствующая блок-схема на рис. 61.

```

* Модель задачи по управлению производством
Nowon  STORAGE  50          ;Число одноврем. работающих машин
Men     STORAGE  3          ;Количество ремонтных рабочих
Machines EQU    53          ;Общее количество машин
*
      Сегмент циркуляции машин
      GENERATE  ,,Machines ;Ввод в систему транзактов-машин
Backplace ENTER   Nowon    ;Машина поступает в работу
      ADVANCE  157,25      ;Машина работает
      LEAVE   Nowon       ;Машина ломается и вых. из строя
      ENTER   Men         ;Занятие рабочего
      ADVANCE  7,3         ;Ремонт машины
      LEAVE   Men         ;Освобождение рабочего
      TRANSFER ,Backplace ;Возврат в оч. резерв. машин
*
      Сегмент таймера
      GENERATE 6240        ;Приход таймера
      TERMINATE 1          ;Окончание моделирования
  
```

Рис. 60. Программа примера 4.5

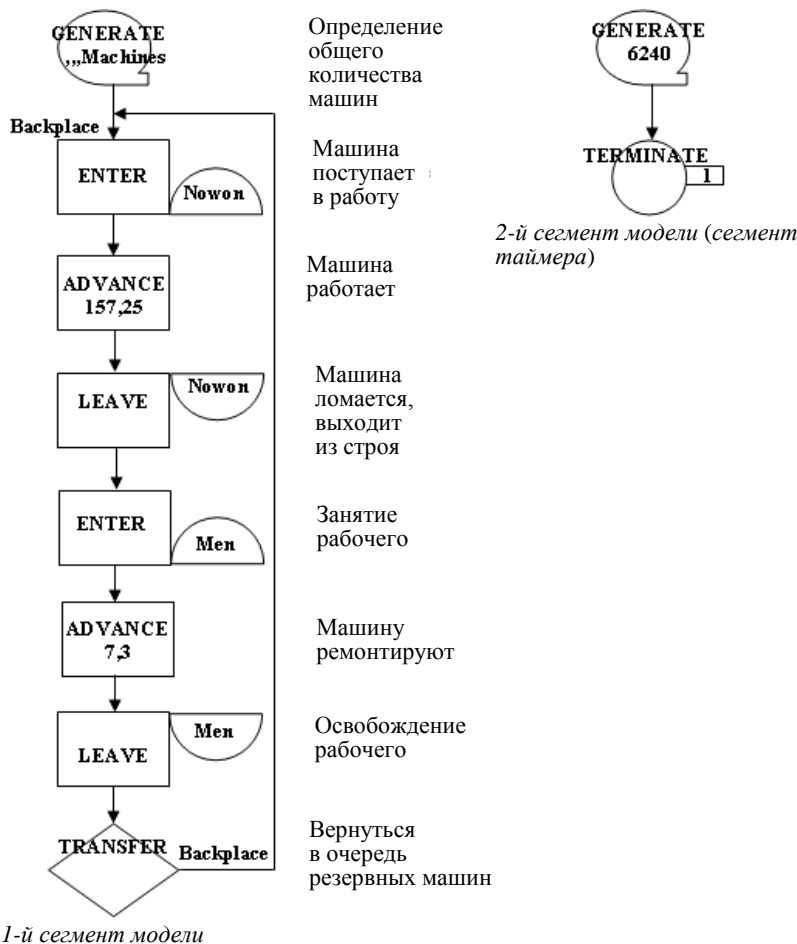


Рис. 61. Блок-схема примера 4.5

Статистика для случая «трое рабочих – три арендуемые машины» показана на рис. 62. Построим функцию зависимости дневных расходов (F) от числа рабочих (x) и количества арендуемых машин (y). Оплата рабочих в ремонтной мастерской за день составит 3,75 долл. США в час \cdot 8 ч в день $\cdot x$. Аренда машин обойдется в 30 долл. США в день $\cdot y$. Потери от простоя машин равны 20 долл. США в час \cdot 8 ч в день \cdot среднее число недостающих машин. Если K – это коэффициент использования устройства Nowon (доля каналов, занятых обслуживанием), то $1 - K$ – это доля свободных каналов. Умножив их на количество каналов памяти Nowon (50), получим среднее число недостающих машин. Таким образом, функция дневных расходов имеет следующий вид:

$$F(x, y) = 30x + 30y + 8\,000(1 - K),$$

где K – коэффициент использования памяти (UTIL. в отчете).

Расчет функции дневных расходов для случая трех рабочих и трех арендуемых машин дает:

$$F(3, 3) = 30 \cdot 3 + 30 \cdot 3 + 8\,000(1 - 0,984) = 308.$$

Пример 35.12.1 - REPORT										
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
NOWON	50	2	0	50	1986	1	49.202	0.984	0	0
MEN	3	0	0	3	1936	1	2.173	0.724	0	2

Рис. 62. Отчет по многоканальным устройствам примера 4.5

Задание 4.4. Из папки *Примеры* загрузите модель *Пример 45.gps*. Модифицируйте модель и выполните прогоны для всех остальных пар (x, y), где $x = 3, 4, 5$ – число рабочих; $y = 3, 4, 5$ – количество арендуемых машин. Значение коэффициента использования памяти Nowon в каждом случае занесите в табл. 19. Рассчитайте дневные расходы для каждого случая и занесите в табл. 20. Выберите наилучший вариант организации системы. Программу модели, таблицы и вывод по задаче приведите в тетради для лабораторных работ.

Таблица 19. Коэффициент использования памяти Nowon

Число рабочих	Число арендуемых машин		
	3	4	5
3	0,984		
4			
5			

Таблица 20. Средние дневные расходы на трикотажной фабрике

Число рабочих	Число арендуемых машин		
	3	4	5
3	308		
4			
5			

4.5. Примеры использования блока TRANSFER

Пример 4.6. Собранные телевизионные радиоприемники после сборки проходят серию испытаний на станциях технического контроля. На последней из этих станций проверяют регулировку установки кадров по вертикали. Если оказывается, что функционирование телевизора ненормально, то отбракованный телевизор переправляют в цех наладки, где заменяют блок установки кадров по вертикали. После наладки телевизор возвращают на последнюю станцию контроля и снова проверяют. Телевизионные приемники уходят с последней станции после одной или нескольких проверок в цех упаковки.

Данная ситуация показана на рис. 63, где неперечеркнутыми кружками представлены телевизоры, ожидающие проверки и проверяемые, а перечеркнутыми – телевизоры с нарушенной регулировкой по вертикали. Последние находятся в наладке либо ожидают обслуживания в цехе наладки.



Рис. 63. Функциональная схема примера 4.6

Телевизионные приемники попадают на последнюю станцию с предыдущей каждые (6 ± 2) мин. На станции находятся 2 контролера. Каждому из них требуется на проверку одного телевизора (9 ± 3) мин. Примерно 85% телевизоров проходят проверку успешно и попадают в цех упаковки. Остальные 15% попадают в цех наладки, в котором находится один рабочий – наладчик. Наладка блока регулировки по вертикали занимает (20 ± 5) мин.

Необходимо разработать модель функционирования этого подразделения и с ее помощью оценить, сколько мест на стеллажах необходимо предусмотреть на входе станции контроля и в цехе наладки. Место на стеллажах – это пространство, предназначенное для хранения ожидающих в очереди телевизоров.

Модель функционирования этой системы состоит из двух сегментов. Первый сегмент изображает путь движения транзактов-телевизоров, а второй служит для ограничения времени моделирования (таймер). Транзакты проходят последовательность QUEUE – ENTER – DEPART – ADVANCE – LEAVE, моделирующую станцию контроля. Из блока LEAVE они входят в блок TRANSFER в режиме статистической передачи. Отсюда в 85% случаев они «проваливаются» в блок TERMINATE. Остальные 15% транзактов переходят в последовательность QUEUE – SEIZE – DEPART – ADVANCE – RELEASE, моделирующую станцию наладки. После выхода из блока RELEASE они безусловно передаются в блок QUEUE, связанный со станцией контроля (рис. 64). Контролеры на станции технического контроля моделируются памятью с именем KONT (емкости 2), а наладчик в цехе наладки представлен устройством FIXER. Очередь к контролерам имеет имя AREA1, а очередь к наладчику – AREA2. Счетчики максимального содержимого обеих очередей можно интерпретировать как объем места для хранения телевизоров на стеллажах.

```

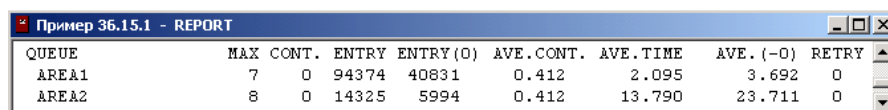
* Модель работы станции технического контроля
KONT STORAGE 2 ;Количество контролеров
* Сегмент моделирования телевизоров
GENERATE 6,2 ;Приход телевизоров из другого цеха
BK QUEUE AREA1 ;Регистрация в очереди на контроль
ENTER KONT ;Занятие контролера
DEPART AREA1 ;Выход из очереди на контроль
ADVANCE 9,3 ;Процесс контроля
LEAVE KONT ;Освобождение контролера
TRANSFER .15,,FIX ;Передача на упаковку или наладку
TERMINATE ;Упаковка телевизоров
FIX QUEUE AREA2 ;Регистрация в очереди к наладчику
SEIZE FIXER ;Занятие наладчика
DEPART AREA2 ;Уход из очереди к наладчику
ADVANCE 20,5 ;Процесс наладки
RELEASE FIXER ;Освобождение наладчика
TRANSFER ,BK ;Возврат на контроль
* Сегмент таймера
GENERATE 480000 ;Таймер приходит через 1000 дней
TERMINATE 1 ;Прекращение моделирования

```

Рис. 64. Модель работы станции технического контроля

За единицу времени принята 1 мин. Время моделирования установим достаточно большое – 1 000 рабочих дней (480 000 мин), чтобы можно было пренебречь временем «разгона» системы, т. е. перехода ее в стационарный режим.

Результаты моделирования показаны на рис. 65. Максимальное число телевизоров в очередях AREA1 и AREA2 равно 7 и 8 соответственно. Из 94 374 телевизоров, прошедших станцию контроля, только 14 325 были отправлены в цех наладки, что составляет около 15,2%.



QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
AREA1	7	0	94374	40831	0.412	2.095	3.692
AREA2	8	0	14325	5994	0.412	13.790	23.711

Рис. 65. Статистические данные об очередях для примера 4.6

Задание 4.5. Из папки *Примеры* загрузите модель *Пример 46.gps*. Модифицируйте эту модель для случая, когда телевизоры поступают в цех контроля каждые (10 ± 2) мин, а вероятность неисправности в них составляет 0,1 (10% телевизоров передаются в цех наладки). Выполните моделирование. Текст программы, результаты и выводы приведите в тетради для лабораторных работ.

Пример 4.7. В парикмахерской с одним мастером имеется только 3 кресла для ожидающих клиентов. Клиенты приходят в парикмахерскую каждые (17 ± 15) мин, но остаются только в том случае, если есть хотя бы одно свободное кресло для ожидания. В противном случае они уходят. Время обслуживания парикмахером составляет (20 ± 10) мин. Определите, какой процент клиентов уходит, не получив обслуживания, т. е. вероятность отказа в обслуживании.

Программа работы парикмахерской на языке GPSS приведена на рис. 66. Многоканальное устройство SEATS символизирует очередь в парикмахерской. После входа в модель транзакт поступает на блок TRANSFER. Параметр В этого блока задан по умолчанию – это следующий за ним блок, поэтому сначала система пытается продвинуть транзакт в блок ENTER SEATS. Если есть свободные места для ожидания, т. е. свободные каналы памяти SEATS, то это ему удается. В противном случае транзакт передается по метке BYBYE на блок TERMINATE, где он уничтожается, имитируя уход клиента, получившего отказ в обслуживании. Моделирование выполняется в течение 8-часового рабочего дня. За единицу времени принята 1 мин.

```

SEATS STORAGE 3
* 1-й сегмент (клиенты)
GENERATE 17,15 ;Приход клиентов
TRANSFER BOTH,,BYBYE ;Проверка наличия свобод. мест
ENTER SEATS ;Занятие стула для ожидания
SEIZE MASTER ;Занятие парикмахера
LEAVE SEATS ;Освобождение стула
ADVANCE 20,10 ;Обслуживание у парикмахера
RELEASE MASTER ;Освобождение парикмахера
TERMINATE ;Уход обслуж. клиентов
BYBYE TERMINATE ;Уход необслуж. клиентов
* 2-й сегмент модели (таймер)
GENERATE 480 ;Приход таймера через 8 ч
TERMINATE 1 ;Окончание моделирования

```

Рис. 66. Модель работы парикмахерской с ограниченной очередью

Результаты моделирования приведены на рис. 67. Из них можно определить, что за время моделирования 29 клиентов вошли в парикмахерскую (поле ENTRY COUNT для блока GENERATE), а 24 из них прошли блок ENTER SEATS (остались для ожидания). Пять транзактов получили отказ в обслуживании (поле ENTRY COUNT для блока с меткой BYBYE). Таким образом, оценка вероятности отказа в обслуживании составляет $5 : 29 = 0,17$, следовательно около 83% пришедших в парикмахерскую клиентов будут обслужены.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT
	1	GENERATE	29	0
	2	TRANSFER	29	0
	3	ENTER	24	1
	4	SEIZE	23	0
	5	LEAVE	23	0
	6	ADVANCE	23	1
	7	RELEASE	22	0
	8	TERMINATE	22	0
BYBYE	9	TERMINATE	5	0
	10	GENERATE	1	0
	11	TERMINATE	1	0

FACILITY	ENTRIES	UTIL.	AVE. TIME AVAIL.	OWNER	PEND	INTEN
MASTER	23	0.943	19.684	1	29	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES AVL.	AVE.C.	UTIL.
SEATS	3	2	0	3	24	1	1.733

Рис. 67. Статистические результаты моделирования парикмахерской с ограниченной очередью

Задание 4.6. Из папки *Примеры* загрузите модель *Пример 47.gps*. Добавьте к этой модели чистильщика сапог. После того, как стрижка заканчивается, 60% клиентов могут захотеть воспользоваться его услугами. Однако никто из них не согласен ждать, если чистильщик занят. Время чистки сапог занимает (20 ± 5) мин. Определите, сколько клиентов в течение рабочего дня почистили сапоги, а сколько хотели, но не смогли этого сделать.

4.6. Задания для самостоятельной работы

Вариант 1

Рабочие приходят на склад каждые (300 ± 250) с. Здесь они получают детали для неисправных станков. Кладовщику требуется (280 ± 150) с на поиск необходимой детали для одного рабочего.

Напишите модель на GPSS для этого случая, выполните моделирование в интервале 8 ч модельного времени. Пусть стоимость потерь из-за поломки станка и простоя рабочего в очереди составляет 0,5 цента в секунду (18 долл. США в час). Каков в этом случае ущерб организации в течение 8-часового рабочего дня в модели?

Предположим, что кладовщик получает 4 долл. США в час. Он может быть заменен другим кладовщиком, получающим 4,5 долл. США в час, но зато выполняющим заявки рабочих за (280 ± 50) с. Выполните моделирование и рассчитайте ущерб от простоев рабочих в этом случае. Укажите, что лучше: оставить старого кладовщика или принять на работу нового.

Вариант 2

Рабочие трех типов приходят на склад за запасными частями. Интервалы времени их прихода и времени обслуживания показаны в табл. 21. На складе работает только один кладовщик. Выполните следующие действия:

1. Напишите на GPSS модель работы такого склада. Затем выполните моделирование в течение одного рабочего дня (8 ч). Модель должна обеспечивать раздельный сбор статистических данных по каждому типу рабочих. Сопоставьте результаты, полученные для очередей трех типов.

2. Модифицируйте модель, чтобы реализовать дисциплину обслуживания «ближайшая операция – кратчайшая». Сопоставьте результаты моделирования с результатами первого пункта, в котором склад работает по принципу «первым пришел – первым обслужен».

Таблица 21. Данные о приходе и обслуживании различных типов рабочих

Типы рабочих	Распределение интервалов времени прихода, мин	Распределение интервалов времени обслуживания, мин
1-й	30 ± 10	12 ± 5
2-й	20 ± 8	6 ± 3
3-й	15 ± 5	3 ± 1

Вариант 3

На станции автосервиса последовательно производится две операции. На каждой из них задействован один рабочий. Время выполнения первой операции (15 ± 7) мин, а второй (30 ± 10) мин. Автомобили прибывают в мастерскую каждые (20 ± 10) мин. Промоделируйте эту систему в течение одного дня (8 ч). Определите максимальное количество автомобилей в очереди на первую и вторую операции, а также коэффициенты загрузки каждого рабочего.

Вариант 4

В авторемонтной мастерской имеется одна полировочная машина. Для полировки определенной детали мотора автомобиля необходимо выполнить следующие этапы:

1. Вынуть деталь ($12 \text{ мин} \pm 3 \text{ мин}$).
2. Установить ее в полировочной машине ($10 \text{ мин} \pm 4 \text{ мин}$).
3. Выполнить первую фазу полировки ($80 \text{ мин} \pm 20 \text{ мин}$).
4. Повернуть деталь в машине для продолжения полировки ($15 \text{ мин} \pm 7 \text{ мин}$).
5. Произвести вторую фазу полировки ($110 \text{ мин} \pm 30 \text{ мин}$).
6. Достать отполированную деталь из машины ($10 \text{ мин} \pm 4 \text{ мин}$).
7. Установить деталь на прежнее место ($12 \text{ мин} \pm 3 \text{ мин}$) и перейти к следующей детали.

Деталь слишком тяжела для того, чтобы ее мог поднять один оператор полировочной машины, поэтому требуется подъемный кран, в частности на первом, втором, четвертом, шестом и седьмом этапах. В авторемонтной мастерской имеется только один подъемный кран, который используется не только для полировки, но и для других целей. Для других видов работ кран может потребоваться через каждые (39 ± 10) мин. Время, на которое забирают кран, равно (25 ± 10) мин.

Постройте модель такой системы. Обеспечьте сбор данных о времени ожидания оператором полировочной машины освобождения крана. Разделите сбор данных об ожидании на четвертом и шестом этапах (предполагается, что если на шестом этапе кран получен, то оператор его уже не отпускает до завершения второго этапа). Соберите также данные об ожидании крана для других видов работ.

Моделирование проведите для 400 ч модельного времени. Сравните значения трех указанных видов данных по каждой из следующих дисциплин обслуживания: первым пришел – первым обслужен; оператор полировочной машины имеет наивысший приоритет при использовании крана.

Вариант 5

В парикмахерскую, в которой работают 2 парикмахера и имеются 2 кресла, клиенты приходят каждые (9 ± 4) мин. Оба парикмахера затрачивают по (16 ± 4) мин на обслуживание одного клиента. Напишите на GPSS модель, собирающую статистику об очереди ожидающих клиентов. Промоделируйте работу системы в течение одного дня. Пусть парикмахерская открывается в 9 ч и закрывается в 17 ч, а парикмахеры работают без перерыва на обед.

Вариант 6

Напишите модель, описывающую работу системы обслуживания, показанную на рис. 68. На первую станцию обслуживания требования (заявки) приходят каждые (115 ± 30) с. Время обслуживания на первой и второй станциях равно (335 ± 60) с и (110 ± 25) с соответственно. Предполагается, что между двумя станциями существует неограниченная очередь.

Используйте модель для определения среднего значения длин двух очередей (моделирование проведите в течение 10 дней, система работает 24 ч в сутки).

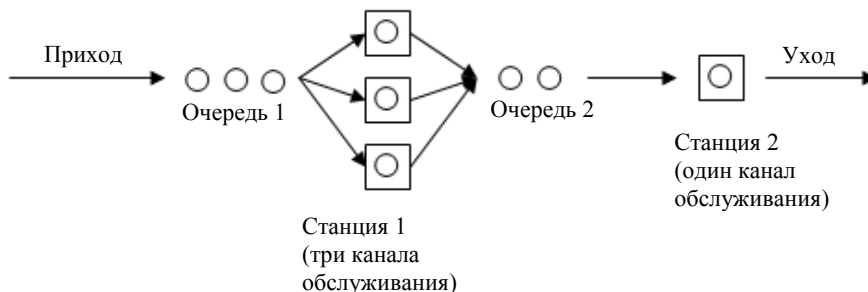


Рис. 68. Модель системы последовательного обслуживания

Вариант 7

Корабли двух типов прибывают в порт, где происходит их разгрузка. В порту имеются 2 буксира, обеспечивающих ввод и вывод кораблей из порта. Первый тип кораблей включает корабли малой тоннажности и требует при входе и выходе использования одного буксира. Второй тип кораблей имеет больший размер, и для их ввода и вывода из порта требуются 2 буксира. Из-за различия размеров кораблей необходимы причалы различного размера, кроме того, корабли имеют разное время разгрузки (погрузки). Данные по этой задаче приведены в табл. 22.

Таблица 22. Данные по кораблям разных типов

Показатели	Типы кораблей	
	1	2
Интервал прибытия, мин	130 ± 30	390 ± 60
Время входа в порт, мин	30 ± 7	45 ± 12
Число доступных причалов	6	3
Время погрузки (разгрузки), ч	12 ± 2	18 ± 4
Интервалы выхода из порта, мин	20 ± 5	35 ± 10

Постройте модель системы, в которой можно оценивать время ожидания кораблями каждого типа входа в порт. Среднее время ожидания включает время освобождения причалов и (или) ожидания буксира. Корабль, ожидающий освобождения причала, не обслуживается буксиром до тех пор, пока не будет предоставлен нужный причал. Более того, корабли второго типа не занимают буксир до тех пор, пока не будут доступны оба буксира. Время моделирования 30 дней круглосуточной работы порта.

Если ожидание кораблями первого и второго типов стоит 350 и 500 долл. США в час соответственно, а стоимость эксплуатации буксира равна 250 долл. США в день, укажите, следует ли воспользоваться третьим добавочным буксиром в порту.

Вариант 8

На таможенном пункте находятся 4 кассы. Каждая из них может обслужить автомобиль за (15 ± 3) мин. Автомобили прибывают в среднем через каждые (3 ± 1) мин.

Промоделируйте эту ситуацию в течение 8 ч. Определите максимальное и среднее число автомобилей, ожидающих в очереди на таможенном пункте. Руководство таможней желает, чтобы среднее время ожидания в очереди было не более 5 мин. Укажите, выполняется ли это условие. Определите, какое количество касс нужно иметь, чтобы оно было выполнено.

Вариант 9

В мастерской по ремонту аппаратуры работают 3 опытных мастера. В среднем телевизоры прибывают в ремонт через (48 ± 10) мин. Каждый из мастеров тратит на ремонт одного телевизора (192 ± 30) мин. Промоделируйте работу системы в течение одной рабочей недели (5 дней по 8 ч). Оцените загруженность мастеров работой и среднее время ожидания ремонта одним телевизором.

Вариант 10

На заправочной станции имеются 3 заправочные колонки. Заправка одного легкового автомобиля длится (120 ± 30) с. Автомобили, нуждающиеся в заправке бензином, прибывают на АЗС каждые (60 ± 10) с. Кроме того, через (330 ± 40) с на АЗС прибывают грузовые машины, которые обслуживаются вне очереди в течение (330 ± 60) с. Определите среднее время пребывания в очереди легкового автомобиля, промоделировав работу системы в течение трех суток и учитывая, что автозаправка открыта 24 ч в сутки.

Вариант 11

В примере моделирования 4.4 предполагалось, что каждый сборщик начинает работу с фазы сборки детали. Измените модель (для четырех сборщиков) таким образом, чтобы она удовлетворяла следующим начальным условиям: один из сборщиков должен начать сборку, второму осталось 10 мин до завершения сборки, третий через 3 мин кончает обжиг, а четвертый ждет возможности занять печь.

Вариант 12

В примере моделирования 4.6 было принято нереальное предположение о том, что даже после того, как телевизор прошел наладку, вероятность его отбраковки все равно остается 0,15. Измените модель так, чтобы после одного или более прохождений через цех наладки вероятность отбраковки становилась 0,03. Как изменятся максимальные значения длин очередей?

Вариант 13

В мастерской работают 2 парикмахера, для ожидания клиентов в ней имеется 3 кресла. Время обработки клиента для первого парикмахера равно (13 ± 3) мин, а для второго – (15 ± 4) мин. Клиенты приходят в парикмахерскую через (6 ± 3) мин, но остаются в ней только в том случае, если есть свободные места для ожидания. Определите характеристики работы системы, в том числе количество клиентов, получивших отказ.

Вариант 14

Добавьте в модель примера 4.7 следующее условие: 40% клиентов, ушедших из парикмахерской ввиду нехватки мест в очереди, через (15 ± 5) мин возвращаются. Если и на этот раз приход безуспешен, они уходят окончательно. Уточните, уменьшает ли это условие количество отказов.

Вариант 15

Измените модель примера 4.7 следующим образом: около 20% клиентов, пришедших в парикмахерскую, остаются только в том случае, если их сразу могут обслужить. Остальные присоединяются к очереди, если в ней есть свободные места. Укажите, уменьшает ли это условие нагрузку мастера.

Вариант 16

Торговая фирма планирует принимать заказы клиентов на приобретение товаров по телефону, для чего устанавливается мини-АТС с тремя телефонными аппаратами. Если заказ поступает, когда все линии заняты, то клиент получает отказ. Предположительно заявки будут поступать с интервалом (4 ± 2) мин. Длительность оформления заказа составляет (8 ± 2) мин. Промоделируйте работу такой системы в течение 8-часового рабочего дня. Укажите, какой процент заявок будет получать отказ.

Контрольные вопросы

1. Возможно ли использование одного и того же имени устройства или очереди в различных сегментах?
2. Как моделируется отдельный сбор статистики по транзактам различного типа: для очереди и для устройства?
3. Что такое приоритет транзакта? Как он задается? Каким образом приоритет транзакта может быть изменен?
4. Какой оператор позволяет изменить маршрут движения транзакта без проверки каких-либо условий?
5. Как работает оператор TRANSFER в режиме BOTH?
6. Как работает оператор TRANSFER в режиме статистической передачи?
7. Каким образом в языке GPSS моделируется многоканальная система массового обслуживания?
8. Что такое емкость памяти?
9. Какие операторы позволяют занять (освободить) канал памяти? Можно ли занять (освободить) сразу несколько каналов?
10. Какие стандартные числовые атрибуты связаны с памятью?
11. Какая статистика собирается по каждой памяти?
12. Какое средство используется для моделирования ограниченной очереди на обслуживание?

ЛАБОРАТОРНАЯ РАБОТА 5 ИСПОЛЬЗОВАНИЕ РАСПРЕДЕЛЕНИЙ ВЕРОЯТНОСТЕЙ В GPSS

5.1. Основные сведения о методах генерации случайных величин в GPSS World

Во всех моделях, рассмотренных ранее, предполагалось равномерное распределение интервалов прихода заявок и времени их обслуживания. Для описания неравномерных распределений разработчик должен воспользоваться функциями языка PLUS, расширяющими возможности языка GPSS. Эти функции являются стандартными в системе GPSS World и могут быть использованы без всяких предварительных описаний.

Всего имеется 24 вероятностных распределения. Рассмотрим некоторые из них.

- Экспоненциальное (показательное) распределение:

$$\text{Real}=\text{Exponential}(RNj,m,s),$$

где RNj – номер генератора равномерных случайных чисел;

m – смещение распределения;

s – масштабный параметр.

Плотность распределения имеет следующий вид:

$$f(x) = \frac{1}{s} e^{-(x-m)/s} \quad (x \geq m). \quad (6)$$

Математическое ожидание $v = m + s$, а дисперсия $D = s^2$.

- Нормальное (гауссово) распределение:

$$\text{Real}=\text{NORMAL}(RNj,m,s).$$

Плотность распределения имеет следующий вид:

$$f(x) = \frac{1}{s\sqrt{2\pi}} e^{-(x-m)^2/2s^2}. \quad (7)$$

Здесь математическое ожидание $\nu = m$ и дисперсия $D = s^2$.

В качестве номера генератора можно использовать любое целое число. При этом начальное значение генератора равно его номеру. Отталкиваясь от этого начального значения, генератор рассчитывает следующее псевдослучайное число через предыдущее. Однако только первые 7 генераторов (с номерами от 1 до 7) могут получить новое начальное значение с помощью оператора RMULT. Например:

RMULT „111

Генератор номер 3 получает начальное значение 111, остальные значения не изменяются.

Любой из генераторов случайных чисел выдает равномерно распределенные случайные величины в интервале от 0,000 000 до 0,999 999 либо от 000 до 999, в зависимости от контекста использования генератора. Функции языка PLUS используют результат, который выдает выбранный генератор, как основу для своих вычислений. Они преобразуют это значение в случайную величину, имеющую заданный закон распределения.

Генераторы, которые используются по умолчанию для каждого вида блоков, требующих розыгрыша случайной величины (GENERATE, ADVANCE и TRANSFER в режиме статистической передачи), задаются командой **Edit/Settings...** на вкладке *Random Numbers*.

5.2. Моделирование пуассоновских потоков событий

Для простейшего (пуассоновского) потока заявок промежуток времени между двумя последовательными приходами заявок имеет экспоненциальный закон распределения. Таким образом, интервал прихода заявок на обслуживание описывается следующей функцией плотности распределения:

$$f_1(t) = \lambda e^{-\lambda t} \quad (t \geq 0), \quad (8)$$

где λ – интенсивность входящего потока заявок, т. е. среднее число заявок, приходящих в единицу времени.

Если в функции (6) положить $m = 0$, а $s = 1/\lambda$ – средний интервал времени между приходом последовательных заявок, то получим требуемую функцию (8). Таким образом, интервал прибытия для пуассоновского потока заявок может быть описан с помощью функции языка PLUS

Exponential(1,0, $T_{приб}$),

где $T_{приб} = 1/\lambda$ – средний интервал времени между поступающими в систему заявками. Здесь использован генератор равномерной случайной величины номер 1, хотя можно использовать и любой другой.

Если продолжительность обслуживания заявки каким-либо каналом распределена экспоненциально, то ее функция плотности распределения имеет вид

$$f_2(t) = \mu e^{-\mu t} \quad (t \geq 0), \quad (9)$$

где μ – интенсивность обслуживания, т. е. среднее число заявок, которое может обслужить канал за единицу времени.

Чтобы получить эту функцию, нужно в формуле (6) положить $m = 0$, а $s = 1/\mu = T_{об}$ – среднее время обслуживания. Таким образом, соответствующая функция языка PLUS имеет следующий вид:

Exponential(1,0, $T_{об}$).

Указанные функции PLUS нужно использовать вместо параметра A в блоках GENERATE и ADVANCE.

Пример 5.1. В кассе метрополитена, продающей карточки на проезд, работают 2 окна. В среднем кассир тратит на обслуживание одного пассажира 0,5 мин, время обслуживания распределено экспоненциально. Входящий поток пассажиров является пуассоновским с интенсивностью 3 человека в минуту. Про моделируйте работу такой системы в течение 80 ч. Найдите среднюю длину очереди, среднее время ожидания в очереди и среднее число каналов, занятых обслуживанием заявок. Сравните эти результаты со значениями соответствующих величин, полученных на аналитической модели.

Модель системы показана на рис. 69.

```

KASSY STORAGE      2
* 1-й сегмент - пассажиры
  GENERATE (Exponential(1,0,3.3)) ;Пуассон. поток заявок
  QUEUE    WKASS                      ;Регистрация в очереди
  ENTER    KASSY                      ;Занятие кассира
  DEPART   WKASS                      ;Рег. выхода из очереди
  ADVANCE  (Exponential(1,0,5)) ;Вр. обл. по эксп. закону
  LEAVE    KASSY                      ;Освобождение кассира
  TERMINATE                      ;Уход пассажира
* 2-й сегмент - таймер
  GENERATE  480000                      ;Таймер через 80 ч
  TERMINATE 1                          ;Прекращение моделир.

```

Рис. 69. Программа модели примера 5.1

Память KASSY имеет емкость 2 и предназначена для моделирования двух открытых окон кассы. Входящий поток заявок является пуассоновским, $\lambda = 3$. Средний интервал времени между приходами пассажиров $T_{приб} = 1/\lambda = 1/3 = 0,33$ мин. Для экспоненциально распределенного времени обслуживания среднее время обслуживания $T_{об} = 0,5$ мин. Примем за единицу времени 0,1 мин. Тогда в блоках GENERATE и ADVANCE данной модели в качестве параметра λ используется функция Exponential(), как показано на рис. 69. Продолжительность моделирования при данной единице времени составит $80 \cdot 60 \cdot 10 = 48\,000$ интервалов по 0,1 мин.

Отчет по результатам моделирования показан на рис. 70. Поле AVE.CONT очереди WKASS содержит среднюю длину очереди, а поле AVE.TIME – среднее время ожидания в очереди. Среднее число занятых каналов обслуживания представлено значением поля AVE.C памяти KASSY. В табл. 23 приведено сравнение результатов, полученных на имитационной и аналитической моделях (для расчета аналитических значений по этой задаче см. практикум [8, с. 38]). Имеющееся несоответствие этих результатов объясняется тем, что результат одного прогона имитационной модели есть результат только одного эксперимента и не имеет той общей значимости, которой обладает аналитическая модель.

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY
WKASS	30	10	14459	5047	2.040	6.771	10.402	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
KASSY	2	0	0	2	14449	1	1.512	0.756	0	10

Рис 70. Результаты моделирования касс метрополитена в течение 80 ч

Таблица 23. Сравнение результатов имитационного и аналитического моделирования

Показатели	Имитационная модель	Аналитическая модель
Средняя длина очереди	2,040	1,931
Среднее время ожидания в очереди, мин	0,68	0,64
Среднее число каналов, занятых обслуживанием	1,512	1,500

Задание 5.1. Из папки *Примеры* загрузите модель *Пример 51.gps*. Модифицируйте модель, чтобы время обслуживания заявок имело нормальный закон распределения со средним значением $m = 0,5$ мин и среднеквадратическим отклонением $s = 0,02$ мин. Сравните результаты прогона этой модели со значениями, полученными для экспоненциального времени обслуживания. В рабочей тетради приведите текст модифицированной модели и таблицу результатов моделирования, аналогичную табл. 23.

5.3. Моделирование эмпирических дискретных функций распределения

Допустим, что закон распределения какой-либо случайной величины нельзя свести к известным стандартным законам (например, к нормальному или экспоненциальному закону). Однако известно, что эта величина является дискретной, т. е. множество ее значений конечно. Кроме того, на основании наблюдений определены относительные частоты ее возможных значений. Например, случайная величина может принимать значения 2, 5, 8, 9 или 12 ед. с относительными частотами, показанными во втором столбце табл. 24.

Таблица 24. Пример дискретной случайной величины

Значения случайной величины	Относительная частота	Суммарная частота	Интервал
2	0,15	0,15	0,00 ÷ 0,15
5	0,20	0,35	0,15 ÷ 0,35
8	0,25	0,60	0,35 ÷ 0,60
9	0,22	0,82	0,60 ÷ 0,82
12	0,18	1,00	0,82 ÷ 1,00

В третьем столбце этой таблицы показана суммарная частота значений случайной величины (функция распределения), а в четвертом – интервал значений суммарной частоты. Предположим теперь, что необходимо разыграть случайное число в соответствии с распределением табл. 24. Сначала разыграем равномерно распределенное случайное число в интервале от 0 до 1. Пусть этим числом будет 0,523 664. Просматривая последний столбец, видим, что это число попадает в третий по счету интервал. Тогда в качестве искомого значения имеем третье значение случайной величины, равное 8.

Таким образом, дискретная случайная величина этого примера может быть промоделирована в виде функции от случайного аргумента RN_j , показанной на рис. 71. Как видно из этого рисунка, каждый интервал включает в себя свою правую границу, но не включает левую. Исключение составляют первый и последний интервалы. Это связано с тем, что значение генератора случайной величины RN_j может быть равно 0, но не может быть равно 1.

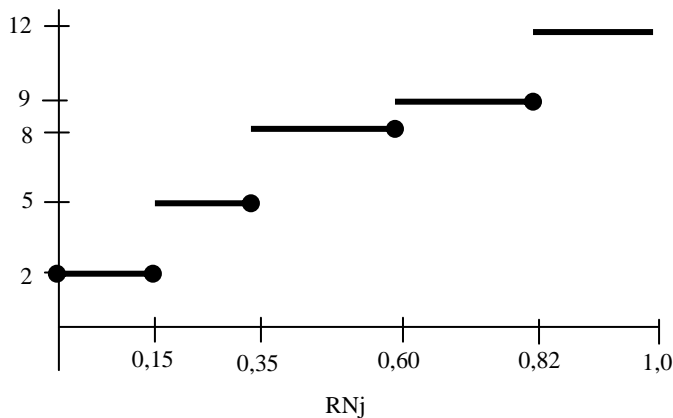


Рис. 71. Графическая интерпретация функции дискретной случайной величины из табл. 24

В языке GPSS описать такую функцию распределения можно с помощью оператора описания FUNCTION:

NAME FUNCTION A,B

Операнды данного оператора имеют следующие значения:

NAME – имя функции, которое должно соответствовать общим правилам составления имен в GPSS;

A – аргумент функции, которым является генератор равномерных случайных чисел в формате RN_j , где j – номер генератора;

B – тип функции (Dm – дискретная, Cm – непрерывная, где m – количество точек табуляции).

За таким описанием в следующей строке следует таблица значений функции в виде совокупности пар *вероятность – значение*, разделенных косыми чертами.

Например, случайная величина из табл. 24 может быть задана в виде следующей функции GPSS с именем PRIHOD:

```
PRIHOD FUNCTION RN1,D5
.15,2/.35,5/.6,8/.82,9/1.0,12
```

Использовать данную функцию можно обращаясь к стандартному числовому атрибуту FN\$PRIHOD.

Задание 5.2. Пусть случайная величина имеет дискретный закон распределения в соответствии с табл. 25. Запишите функцию GPSS, отражающую этот закон распределения. Укажите, как можно использовать данную функцию. Постройте график соответствующей функции случайного аргумента RN_j .

Таблица 25. Закон распределения дискретной случайной величины

Значения случайной величины	6	7	8	9	10
Относительная частота	0,05	0,25	0,40	0,25	0,05

5.4. Моделирование эмпирических непрерывных функций распределения

Рассмотрим непрерывную случайную величину, закон распределения которой неизвестен, а область возможных значений разбита на некоторые интервалы. В результате статистических наблюдений были определены относительные частоты попадания значений этой случайной величины в каждый из интервалов.

Например, диапазон времени, требуемый кассиру банка для обслуживания клиента, в результате наблюдений не стал меньше 15 и больше 90 с, он был разбит на интервалы в 15 с. Относительные частоты попадания времени обслуживания в каждый интервал приведены в табл. 26.

Таблица 26. Статистика времени обслуживания кассиром банка

Интервал времени обслуживания, с	Относительная частота попадания времени обслуживания в интервал	Суммарная частота
$(-\infty, 15)$	0,00	0,00
$[15, 30)$	0,07	0,07
$[30, 45)$	0,25	0,32
$[45, 60)$	0,41	0,73
$[60, 75)$	0,19	0,92
$[75, 90]$	0,08	1,00

Таким образом, 7% обслуживаний были завершены за время от 15 до 30 с (не включая 30 с), а 41% – за время от 45 до 60 с (не включая 60 с) и т. д. Поскольку другая информация отсутствует, предположим, что на каждом интервале в 15 с все значения времени обслуживания равновероятны, имеют равномерное распределение.

Рассчитаем суммарную частоту для каждого интервала времени обслуживания (см. третий столбец табл. 26) и определим непрерывную функцию GPSS, соответствующую этим наблюдениям:

```
SERVE FUNCTION RN2,C6
.0,15/.07,30/.32,45/.73,60/.92,75/1.0,90
```

Описание такой функции отличается лишь наличием символа C вместо D во втором аргументе этого оператора.

При обращении к непрерывной функции GPSS разыгрывается случайное число из диапазона от 0,000 000 до 0,999 999, которое далее используется как аргумент функции. Затем просматривается таблица для определения интервала значений суммарной частоты, на который выпало разыгранное число. Выполняется линейная интерполяция для пары точек, находящихся по краям этого интервала. График функции случайного аргумента RNj, который соответствует непрерывной случайной величине из табл. 26, показан на рис. 72.

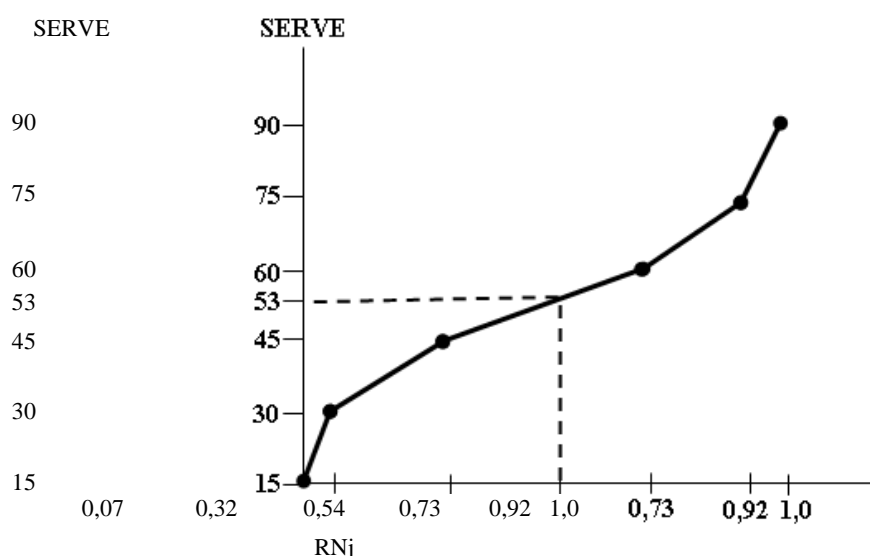


Рис. 72. Графическая интерпретация непрерывной случайной функции

Пусть, например, генератор равномерной случайной величины RN2 дал значение 0,544 281. Это число попадает в интервал от 0,32 до 0,73. Уравнение прямой, соединяющей точки, соответствующие концам этого интервала, следующее:

$$\frac{x - 0,32}{0,73 - 0,32} = \frac{y - 45}{60 - 45}.$$

Для $x = 0,544\ 281$ получаем на этой прямой $y \approx 53$. Таким образом, функция SERVE даст в этом случае значение 53.

Задание 5.3. Пусть время между приходом заявок является непрерывной случайной величиной с неизвестным законом распределения. В результате эксперимента на реальной системе весь диапазон значений этой величины был разделен на 4 интервала, и были измерены относительные частоты попадания значений случайной величины в эти интервалы (табл. 27). Постройте непрерывную функцию GPSS, отражающую этот закон распределения. Запишите оператор создания транзактов, который использует эту функцию. Определите, какое значение выдаст эта функция, если генератор равномерной случайной величины дал значение 0,5.

Таблица 27. Статистика интервалов между приходами заявок

Интервалы времени между заявками, мин	[0, 5)	[5, 12)	[12, 36)	[36, 40)
Относительная частота попадания в интервал	0,11	0,06	0,5	0,33

Ответ: 27,84.

5.5. Определение непрерывных равномерных функций

Ввиду того, что непрерывная функция GPSS задается с помощью линейной интерполяции, для всех точек на одном интервале значений суммарной частоты существует одинаковая вероятность их использования. Это дает простой способ задания непрерывного равномерного распределения в интервале от a до b . Пусть, например, некоторая случайная переменная распределена равномерно и непрерывно в интервале [2, 6). Вероятность того, что значение этой переменной меньше 2, равна 0, а вероятность того, что она меньше 6, равна 1. Эти два значения суммарной вероятности используют для определения непрерывной функции:

```
GOOD FUNCTION RN2,C2
0,2/1,6
```

Графическая интерпретация этой функции приведена на рис. 73. При обращении к функции используется генератор RN2, значения которого применяют при интерполяции. Например, если значением RN2 является 0,650 000, то значением функции будет 4,6.

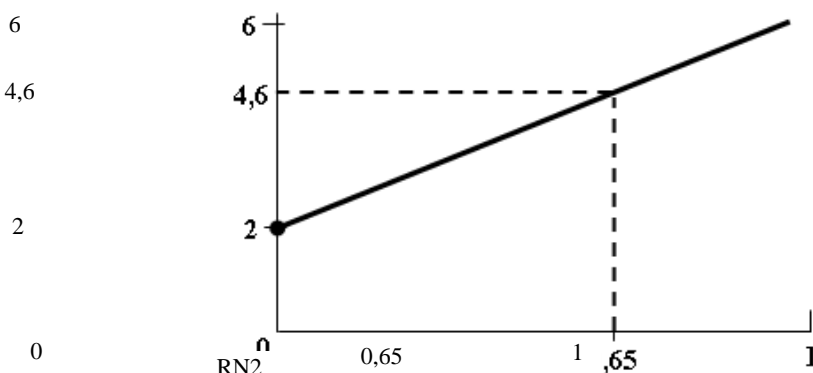


Рис. 73. Графическая интерпретация равномерной непрерывной функции

Таким образом, следующие записи блока GENERATE почти эквивалентны:

```
GENERATE 4,2
GENERATE FN$GOOD
```

Отличие состоит в том, что во втором случае значение интервала времени между приходами, равное 6, не может быть получено никогда (максимальное значение будет равно 5,999 999).

5.6. Задания для самостоятельной работы

Вариант 1

В отделении банка клиентов обслуживают 3 оператора. Среднее время обслуживания одного клиента составляет 12 мин. В среднем за час обращаются 15 клиентов. Если все операторы заняты, клиенты не остаются для ожидания и покидают систему без обслуживания. Считая все потоки в системе пуассоновскими, выполните моделирование этой системы в течении 80 ч и определите вероятность отказа в обслуживании, абсолютную пропускную способность и среднее число занятых каналов. Сравните эти значения с результатами аналитических расчетов.

Вариант 2

В пункте обмена валют работают 2 оператора, каждый из которых обслуживает клиента в среднем за 2,5 мин. По условиям безопасности в помещении пункта может находиться одновременно не более 5 человек, включая обслуживаемых клиентов. Если помещение заполнено, то очередной клиент не становится в очередь, а уходит. В среднем клиенты приходят каждые 2 мин. Считая все потоки событий пуассоновскими, промоделируйте систему в течение 80 ч. Найдите основные характеристики работы обменного пункта ($P_{отк}$, A , K , $L_{оч}$, $T_{оч}$) и сравните их с аналитическими расчетами.

Вариант 3

Информационно-справочное бюро имеет 2 телефона, по которым дает справки о наличии товаров в магазинах. В среднем за минуту поступает 3 запроса, поток запросов на обслуживание является пуассоновским. Время обслуживания каждого запроса имеет экспоненциальный закон распределения со средним значением 40 с. Осуществите моделирование системы в течение 50 ч, определите вероятность отказа абоненту, абсолютную пропускную способность и среднее число занятых каналов обслуживания. Сравните найденные показатели с результатами аналитических расчетов.

Вариант 4

В мастерской по ремонту радиоаппаратуры работают 5 опытных мастеров. В течение дня в ремонт поступают в среднем 10 аппаратов и каждый из мастеров успевает отремонтировать за день около 2,5 аппарата. Все потоки событий пуассоновские. Промоделируйте работу мастерской в течение 10 лет (3 650 рабочих дней). Сравните основные показатели работы системы (K , $L_{оч}$, $T_{оч}$) с результатами аналитических расчетов.

Вариант 5

Фирма «Уют» предлагает своим клиентам услуги по дизайну для дома и офиса. В нормальном режиме каждый час в фирму прибывает в среднем 2,5 клиента, поток прибывающих клиентов пуассоновский. Единственный консультант по дизайну отвечает на вопросы клиентов и дает необходимые рекомендации. Время обслуживания посетителя имеет экспоненциальный закон распределения и составляет в среднем 10 мин. Выполните моделирование в течение 100 ч. Определите среднее время ожидания и среднюю длину очереди, сравните эти показатели с результатами аналитических расчетов.

Вариант 6

Входной поток клиентов в парикмахерскую подчиняется закону Пуассона, его интенсивность составляет 12 человек в час. Посетителей обслуживают 3 мастера. В среднем за час мастер обслуживает 2 человека, время обслуживания распределено экспоненциально. Максимальная очередь 5 человек. Определите основные характеристики работы парикмахерской ($P_{отк}$, A , K , $L_{оч}$, $T_{оч}$), выполнив моделирование в течение 100 ч. Сравните результаты моделирования с результатами аналитических расчетов.

Вариант 7

Автоматическая телефонная станция имеет одну линию, на которую в среднем приходит 0,8 вызова в минуту. Среднее время разговора 1,5 мин. Вызов, пришедший во время разговора, не обслуживается. Считая потоки вызовов пуассоновскими, найдите абсолютную и относительную пропускную способности станции, а также среднее время пребывания заявки в системе. Выполните моделирование в течение 80 ч. Сравните найденные значения показателей с результатами аналитических расчетов.

Вариант 8

На автостанции имеется 5 автоматических камер хранения. Услугами этих камер пользуются в среднем 8 пассажиров в сутки, поток пассажиров пуассоновский. Время хранения багажа распределено экспоненциально, со средним значением 12 ч. Вычислите вероятность отказа пассажирам в обслуживании, промоделировав работу системы в течение 365 сут. Сравните этот показатель с результатом аналитических расчетов. Определите, какое количество камер хранения надо иметь, чтобы эта вероятность не превышала 0,07.

Контрольные вопросы

1. Что обозначает стандартный числовой атрибут RN_j ?
2. Для чего в GPSS используется генератор равномерных случайных чисел? Какие значения он выдает?
3. Как задается начальное значение генератора равномерных случайных чисел? На какие генераторы равномерных случайных чисел может воздействовать оператор RMULT?
4. Каким образом задаются номера генераторов случайных чисел, которые используются в модели по умолчанию для блоков GENERATE, ADVANCE и TRANSFER?
5. Какие функции GPSS нужно использовать для марковских СМО, если средний интервал времени между приходами заявок $T_{приб}$, а среднее время обслуживания $T_{об}$? Напишите их.
6. В чем отличие дискретной случайной величины и непрерывной?
7. Выполните графическую иллюстрацию функции
PP FUNCTION RN3,D4
.3,8/.65,12/.8,18/1.0,25
Как рассчитывается значение этой функции для $RN3 = 0,7$?
8. Выполните графическую иллюстрацию функции
PPC FUNCTION RN3,C4
.3,8/.65,12/.8,18/1.0,25
Как рассчитывается значение этой функции для $RN3 = 0,7$?
9. Запишите равномерную непрерывную функцию в интервале от 100 до 400. Чему равно значение этой функции для $RN_j = 0,3$?

ЛАБОРАТОРНАЯ РАБОТА 6 ПРИЕМЫ ПРОГРАММИРОВАНИЯ МОДЕЛЕЙ (ПРОДОЛЖЕНИЕ)

6.1. Использование стандартных числовых атрибутов

Стандартные числовые атрибуты – это значения системных переменных, которые доступны пользователю в ходе выполнения программы модели. В третьей и четвертой лабораторных работах были приведены некоторые СЧА, которые могут быть использованы в качестве операндов блоков модели, например:

ENTER 3,R3

Транзакт при входе в этот блок занимает все оставшиеся до его входа свободные каналы памяти с номером 3. СЧА R3 обозначает число свободных каналов памяти 3.

DEPART Wait,Q\$Wait

СЧА Q\$Wait обозначает текущее количество транзактов в очереди Wait. Таким образом, когда транзакт проходит этот блок, счетчик очереди Wait уменьшается на число транзактов, находящихся в очереди в данный момент.

Кроме того, СЧА могут быть использованы в качестве аргументов функций. Функции, которые рассматривались в лабораторной работе 5, в качестве аргумента использовали генераторы RN_j . Однако это не обязательно, аргументом функции может быть и другой СЧА. Рассмотрим, например, следующую дискретную функцию:

MAP FUCTION N\$MARY,D4
5,1/8,2/10,5/15,7

Ее аргументом является счетчик входов в блок с именем MARY. При обращении к функции в качестве независимой переменной используется этот счетчик. Значение же функции рассчитывается обычным образом, как это принято для дискретной функции. График функции показан на рис. 74. Когда аргумент функции выходит за пределы, описанные при определении функции, ее значение принимается равным ближайшим описанным значениям. Например, при аргументе, меньшем 5, значение функции равно 1. При аргументе, большем 15, значение функции равно 7.

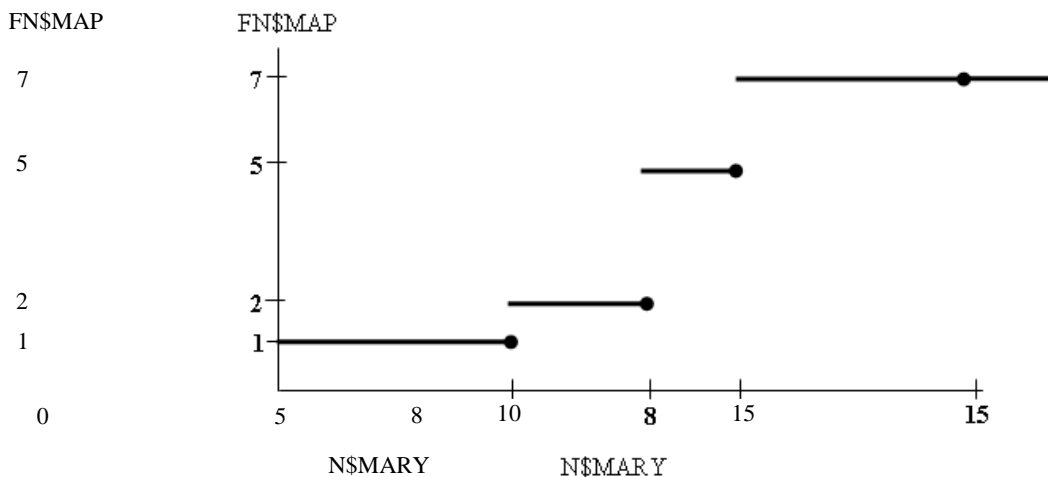


Рис. 74. График дискретной функции, аргументом которой является счетчик входов в блок MARY

Аналогичная непрерывная функция рассчитывается методом линейной интерполяции:

```
MAP FUNCTION N$MARY,C4
5,1/8,2/10,5/15,7
```

График этой функции показан на рис. 75. При выходе за пределы описания аргумента, значение функции принимается равным ближайшему описанному значению.

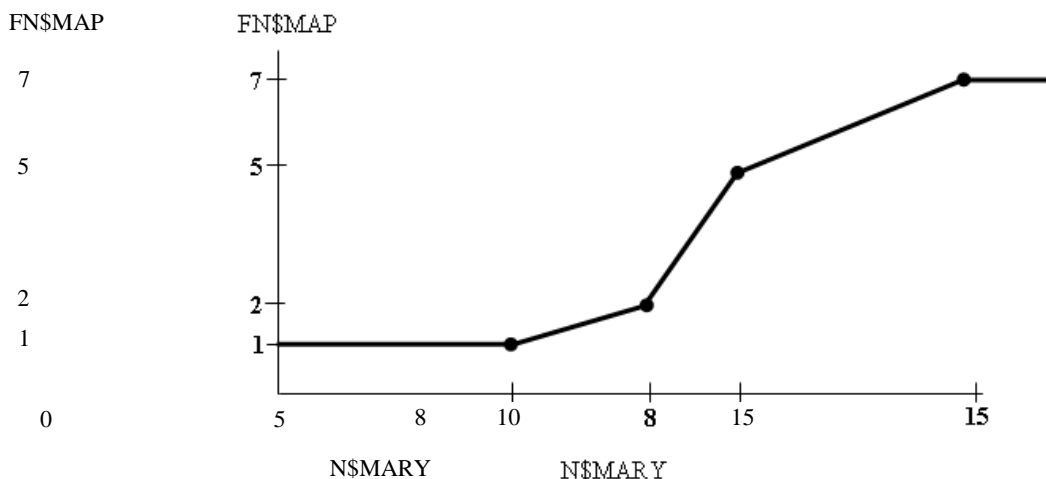


Рис. 75. График непрерывной функции, аргументом которой является счетчик входов в блок MARY

Пример 6.1. В системе массового обслуживания с одним устройством и очередью имеет место пуассоновский входящий поток заявок с интенсивностью 12 приходов в час. Обслуживание является экспоненциальным, но среднее время обслуживания зависит от числа требований, находящихся в очереди к устройству. Эта зависимость представлена в табл. 28.

Таблица 28. Среднее время обслуживания как функция длины очереди

Длина очереди	Среднее время обслуживания, мин
0	5,5
1 или 2	5,0
3,4 или 5	4,5
6 и более	4,0

Необходимо промоделировать систему в течение 100 ч и оценить фактическое среднее время обслуживания. Кроме того, требуется получить ответ на вопрос, справится ли устройство с нагрузкой при увеличении интенсивности входящего потока на один приход в час или же очередь будет бесконечно расти.

Текст программы на языке GPSS приведен на рис. 76. Обслуживающее устройство имеет имя SURVR, а очередь к нему – имя Wait. Функция Mean имеет своим аргументом текущую длину очереди Wait (СЧА

Q\$Wait). Единицей времени является 1 с. Поэтому средний интервал между приходами транзактов в систему равен $60 \cdot 60 : 12 = 300$ с. В функции Mean время также дано в секундах (5,5 мин = 330 с). В блоке ADVANCE используется стандартная функция экспоненциального распределения, а для расчета ее аргумента, представляющего среднее значение интервала обслуживания, используется вложенная функция Mean.

```
Mean FUNCTION Q$Wait,D4 ;Сред. время обслуж.
0,330/2,300/5,270/6,240 ;зависит от длины очереди
* 1-й сегмент - клиенты
GENERATE (Exponential(1,0,300)) ;Приход клиента
QUEUE Wait ;Регистрация в очереди
SEIZE SURVR ;Занятие устройства
DEPART Wait ;Регистр. вых. из очереди
ADVANCE (Exponential(1,0, FN$Mean)) ;Экспон. вр. обслуж.
;зависит от длины очереди
RELEASE SURVR ;Обслуживание устройства
TERMINATE ;Уход из системы
* 2-й сегмент - таймер
GENERATE 360000 ;Таймер через 100 ч
TERMINATE 1 ;Прекращ. моделирования
```

Рис. 76. Программа примера 6.1

Результаты моделирования этой системы приведены на рис. 77. Анализ этих результатов показывает, что фактическое среднее время обслуживания составило 284 с, коэффициент нагрузки устройства равен 0,915, а средняя длина очереди – 3,5 заявки.

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
SURVR	1158	0.915	284.437	1	1159	0	0	0	4

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
WAIT	17	4	1162	112	3.474	1076.340	1191.150

Рис. 77. Результаты моделирования примера 6.1

Задание 6.1. Из папки *Примеры* загрузите модель *Пример 61.gps*. Модифицируйте модель, увеличив интенсивность входящего потока на один приход в час. Получите результаты моделирования и сравните их с результатами примера 6.1. В тетради приведите текст модифицированной модели и таблицу результатов моделирования для двух случаев нагрузки.

6.2. Параметры транзактов и арифметические переменные

С каждым транзактом связан набор параметров. Количество параметров в GPSS World не ограничено. Параметр имеет номер – положительное целое число, или символическое имя. Значение параметра активного транзакта можно получить с помощью стандартного числового атрибута P_j (или $P\$имя$), где j – номер (имя) параметра. Значения параметров активного транзакта можно использовать в качестве аргументов блоков, функций и т. д.

Присвоить значение параметру активного транзакта можно с помощью блока ASSIGN, который имеет формат

ASSIGN A,B,C

Операнды данного блока имеют следующие значения:

- A – номер или имя параметра активного транзакта, за которым может следовать знак «+» или «-»;
- B – присваиваемое значение параметра;
- C – номер функции-модификатора значения, указанного в поле B.

Если до блока ASSIGN параметр с таким номером или именем не существовал, то он создается. Графическое изображение блока ASSIGN приведено на рис. 78.

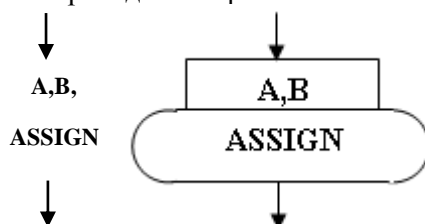


Рис. 78. Графическое изображение блока ASSIGN

Примеры

ASSIGN 2000,150.6

Параметру активного транзакта номер 2 000 присваивается значение 150.6.

ASSIGN TEXT,“Привет”

Параметру с именем TEXT присваивается значение “Привет”.

ASSIGN 2000+,-3

Параметр номер 2 000 увеличивается на -3. Если до этого блока параметр не существовал, то он создается, и ему присваивается начальное значение 0. Таким образом, после прохождения этого блока транзакт будет иметь параметр с номером 2 000, значение которого равно -3:

ASSIGN 2000-,-3

Параметр номер 2 000 уменьшается на -3 (фактически увеличивается на 3).

В случае использования аргумента C он представляет собой номер (имя) функции, без предшествующего FN. Вычисляется значение этой функции и умножается на значение аргумента B. Полученный результат присваивается параметру с номером A.

Пример

ASSIGN 4,2,1

Значение функции FN1 умножается на 2, и результат этих вычислений присваивается параметру 4 активного транзакта.

В языке GPSS используется понятие переменной, которая служит инструментом расчета значений выражений, т. е. выступает в качестве математической функции. Для описания переменных существуют операторы описания VARIABLE (переменная с фиксированной точкой), FVARIABLE (переменная с плавающей точкой) и BVARIABLE (булевская переменная):

NAME VARIABLE A

В этой записи:

NAME – имя переменной;

A – выражение, которое может включать в себя числовые константы, СЧА, знаки операций и математические функции языка PLUS.

Для обращения к значению переменной используется стандартный числовой атрибут V\$NAME.

Примеры

JJK2 VARIABLE 10#(11/3)

Для обращения к такой переменной используется СЧА V\$JJK2, значение которого равно 36,667 (знак «#» в GPSS используется в качестве знака умножения).

Varl VARIABLE 5#LOG(Q\$Wait)

При обращении к такой переменной V\$Varl берется текущее значение длины очереди Wait, вычисляется натуральный логарифм этого числа и умножается на 5. Результат вычислений является значением переменной.

Пример 6.2. Модель продовольственного магазина. Небольшой продовольственный магазин состоит из 3 прилавков и 1 кассы, которая находится на выходе из магазина. Входной поток покупателей в магазин имеет пуассоновский характер, причем среднее значение интервала прихода составляет 75 с. Войдя в магазин, каждый покупатель берет корзину и может обойти один или несколько прилавков, отбирая продукты. Вероятность обхода конкретного прилавка показана в табл. 29. Время, требуемое для обхода прилавка, и число покупок, выбранных у прилавка, распределены равномерно. Эти данные также приведены в табл. 29.

Таблица 29. Характеристики прилавков продовольственного магазина

Прилавки	Вероятность выполнения покупок	Время обхода прилавка, с	Число покупок, совершаемых у прилавка, шт.
1-й	0,75	120 ± 60	3 ± 1
2-й	0,55	150 ± 30	4 ± 1
3-й	0,82	120 ± 45	5 ± 1

После того, как товар отобран, покупатель становится в очередь к кассе. Уже стоя в очереди, покупатель может захотеть сделать еще (2 ± 1) покупки. Время обслуживания покупателя в кассе пропорционально числу сделанных покупок, на одну покупку уходит 3 с проверки. После оплаты продуктов покупатель оставляет корзину и уходит.

Необходимо построить модель, описывающую процесс покупок в магазине, определить нагрузку кассира и максимальную длину очереди перед кассой. Временем моделирования является 8-часовой рабочий день. Считая, что число корзин неограничено, определите максимальное количество корзин, находящихся у покупателей одновременно.

Программа модели магазина приведена на рис. 79.

Единицей времени в модели является 1 с. Транзакт-таймер приходит в момент времени $8 \cdot 60 \cdot 60 = 28\,800$ с, которые соответствуют 8 ч.

Первый сегмент модели имитирует покупателя. При входе этого транзакта в модель он сразу же попадает в многоканальное устройство (память) Carts. Емкость этой памяти заведомо очень велика, поэтому вход транзактов в нее никогда не ограничивается и очередь не образуется. Эта память требуется только для сбора статистики по использованию корзин.

Подход к каждому прилавку моделируется с использованием блока TRANSFER в режиме статистической передачи. Если покупатель решает не делать покупок у одного из прилавков, то транзакт, который его моделирует, переходит к новому блоку TRANSFER, находящемуся перед очередным прилавком. Если же покупатель остается для выполнения покупок, то его обход прилавка моделируется задержкой на случайное время, указанное в табл. 29. Затем, согласно дискретной функции (Pok1, Pok2 или Pok3), разыгрывается случайное количество покупок у этого прилавка и добавляется к первому параметру транзакта. Этот параметр служит для регистрации общего числа покупок.

```
Carts STORAGE 1000 ;Очень большое кол-во корзин
Pok1 FUNCTION RN1,D3 ;Кол-во покупок за прилавком 1
0.333,2/0.667,3/1,4
Pok2 FUNCTION RN1,D3 ;Кол-во покупок за прилавком 2
0.333,3/0.667,4/1,5
Pok3 FUNCTION RN1,D3 ;Кол-во покупок за прилавком 3
0.333,4/0.667,5/1,6
Impul FUNCTION RN1,D3 ;Кол-во дополнительных покупок
0.333,1/0.667,2/1,3
Prov VARIABLE P1#3 ;Расчет времени обслуж. кассиром
* 1-й сегмент - покупатель
GENERATE (Exponential(1,0,75)) ;Приход покупателя
ENTER Carts ;Занятие корзины
TRANSFER .25,,Try2 ;Нужны ли покупки за прилавком 1?
ADVANCE 120,60 ;Обход прилавка 1
ASSIGN 1,FN$Pok1 ;В парам. 1 записать к-во покупок
Try2 TRANSFER .45,,Try3 ;Нужны ли покупки за прилавком 2?
ADVANCE 150,30 ;Обход прилавка 2
ASSIGN 1+,FN$Pok2 ;Добавить покупки у прилавка 2
Try3 TRANSFER .18,,Out ;Нужны ли покупки у прилавка 3?
ADVANCE 120,45 ;Обход прилавка 3
ASSIGN 1+,FN$Pok3 ;Добавить покупки у прилавка 3
Out QUEUE Check ;Встать в очередь к кассе
ASSIGN 1+,FN$Impul ;Сделать дополнит. покупки
SEIZE Checker ;Занять кассира
DEPART Check ;Покинуть очередь к кассе
ADVANCE V$Prov ;Проверка кассиром
RELEASE Checker ;Освободить кассира
LEAVE Carts ;Вернуть корзину
TERMINATE ;Покинуть магазин
* 2-й сегмент - таймер
GENERATE 28800 ;Приход таймера через 8 ч
TERMINATE 1 ;Завершение прогона
```

Рис. 79. Программа модели продовольственного магазина

После обхода прилавков транзакт-покупатель регистрируется в очереди к кассе Check, после чего разыгрывается случайное число дополнительных покупок согласно функции Impul, которое прибавляется к первому параметру. Время обслуживания у кассира на устройстве Checker моделируется задержкой на время, пропорциональное числу покупок. Для этого используется переменная Prov, значение которой рассчитывается как произведение значения первого параметра транзакта на число 3.

Результаты прогона модели показаны на рис. 80.

Пример 52.6.1 - REPORT										
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
CHECKER	342	0.374	31.491	1		0	0	0	0	0
QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY		
CHECK	3	0	342	198	0.127	10.696	25.403	0		
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE. C.	UTIL.	RETRY	DELAY
CARTS	1000	997	0	11	345	1	3.707	0.004	0	0

Рис. 80. Содержимое отчета по модели продовольственного магазина

Коэффициент нагрузки кассира составляет 0,374. Статистика очереди показывает, что обслуживания перед кассой ожидают в среднем 0,127 человека. Те, кому пришлось ждать в очереди, в среднем находились в ней 25 с. Максимальное количество покупателей в очереди равно 3. Статистические данные по памяти Carts дают возможность определить максимальное количество одновременно используемых корзин – 11.

Задание 6.2. Из папки *Примеры* загрузите модель *Пример 62.gps*. Модифицируйте модель для случая, когда кассир является неопытным и тратит на каждую покупку 5 с плюс 10 с на подсчет сдачи для каждого покупателя. Приведите в отчете и поясните статистические результаты прогона этой модели.

6.3. Изменение приоритета транзакта

При входе транзакта в модель уровень его приоритета определяется значением операнда Е соответствующего блока GENERATE. В процессе моделирования можно динамически изменять приоритет транзакта. Для этого служит блок

PRIORITY A,B

Значения его операндов следующие:

A – присваиваемое значение приоритета активного транзакта;

B – опция буферирования, которая может быть опущена или иметь значение BU, тогда транзакт помещается в цепи текущих событий за транзактами с таким же значением приоритета, в противном случае он помещается перед ними.

Графическое изображение блока PRIORITY показано на рис. 81.

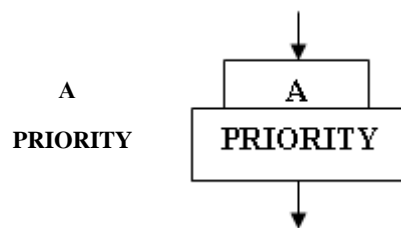


Рис. 81. Графическое изображение блока PRIORITY

Уровень приоритета активного транзакта является стандартным числовым атрибутом GPSS с именем PR. Этот атрибут может быть использован в качестве операнда блока, аргумента функции и т. д.

Пример 6.3. После завершения литья в литейном цехе отливка переносится в другой цех, где происходит ее окончательная обработка. Для выполнения этой обработки используется агрегат доводки, которым управляет один оператор. Доводка состоит из пяти этапов:

1. Процедура 1, распределение времени выполнения которой приведено в табл. 30.
2. Поворот отливки, на который требуется (15 ± 5) мин.
3. Процедура 2, распределение времени выполнения которой приведено в табл. 31.
4. Последовательность действий, занимающая (30 ± 5) мин, а именно:
 - выгрузка обработанной отливки из агрегата;
 - укладка обработанной отливки;
 - подъем следующей отливки из штабеля;
 - загрузка ее в агрегат.
5. Возврат к первой процедуре.

Таблица 30. Распределение времени, необходимого для выполнения 1-й процедуры

Время выполнения 1-й процедуры, мин	Суммарная частота
Менее 60	0,00
70	0,12
80	0,48
90	0,83
100	1,00

Таблица 31. Распределение времени, необходимого для выполнения 2-й процедуры

Время выполнения 2-й процедуры, мин	Суммарная частота
Менее 80	0,00
90	0,24
100	0,73
90	0,83
110	1,00

Отливка довольно тяжела, поэтому для выполнения второго и четвертого этапов требуется подъемный кран. Поскольку первый и третий этапы требуют много времени, кран должны обслуживать несколько агрегатов, чтобы простой его были меньше. Однако при большом количестве агрегатов появляются и увеличиваются простои операторов этих агрегатов по причине ожидания освобождения крана. Требуется определить, какое число агрегатов целесообразно обслуживать одним краном для достижения баланса в использовании оборудования.

Следует учитывать, что в работе крана могут возникать конфликтные ситуации. В некоторый момент времени один или более агрегатов могут потребовать кран для выполнения второго этапа, а другие агрегаты требуют кран для выполнения четвертого этапа. В этой ситуации второй этап должен иметь преимущество перед четвертым, так как он требует меньше времени (принцип «ближайшая операция – кратчайшая»).

Моделирование необходимо выполнить в течение 40-часовой рабочей недели.

За единицу времени примем 1 мин. Таймер окончания моделирования приходит через $40 \cdot 60 = 2\,400$ ед. модельного времени.

Транзакт в модели соответствует оператору агрегата доводки. Предположим, что в цехе работает 7 агрегатов доводки, а следовательно, и 7 операторов. Поэтому в модель вводятся 7 транзактов оператором GENERATE ...,7,1. Приоритет этих транзактов при входе в модель равен единице. Работа оператора представляет собой цикл повторения этапов от 1 до 4, причем каждый из них моделируется задержкой на время его выполнения.

Кран имитируется устройством KRAN, который оператор агрегата должен занять перед выполнением второго и четвертого этапов. Очевидно, что запросы на кран образуют очередь, но статистика по этой очереди в данной задаче не требуется, поэтому операторы QUEUE – DEPART не используются. Чтобы обеспечить преимущество второго этапа перед четвертым при занятии крана, будем динамически изменять приоритет транзакта, задавая его равным единице перед вторым этапом и равным нулю перед четвертым. Предварительно текст программы будет такой, как показано на рис. 82.

```

PROC1  FUNCTION  RN1,C5      ;Время выполнения процедуры 1
0,60/.12,70/.48,80/.83,90/1,100
PROC2  FUNCTION  RN1,C4      ;Время выполнения процедуры 2
0,80/.24,90/0.73,100/1,110
* Сегмент операторов агрегата доводки
  GENERATE      ,,,7,1      ;Семь операторов с приоритетом 1
CYCLE  ADVANCE  FN$PROC1    ;Выполнение процедуры 1
  SEIZE         KRAN        ;Получение крана
  ADVANCE       15,5        ;Поворот отливки
  RELEASE       KRAN        ;Освобождение крана
  PRIORITY      0           ;Приор. 0 для след. занят. крана
  ADVANCE       FN$PROC2    ;Выполнение процедуры 2
  SEIZE         KRAN        ;Получение крана
  ADVANCE       30,5        ;Выгрузка, укладка, загрузка
  RELEASE       KRAN        ;Освобождение крана
  PRIORITY      1           ;Приор. 1 для след. занят. крана
  TRANSFER      ,CYCLE      ;Переход к следующей отливке
* Сегмент таймера
  GENERATE      2400
  TERMINATE     1

```

Рис. 82. Первый вариант модели примера 6.3

Поскольку по устройству KRAN в модели автоматически собирается статистика моделирования, в отчете будет получена нагрузка крана. А вот нагрузка операторов агрегатов доводки в такой модели не фиксируется. Чтобы организовать сбор статистики по операторам, используется память (многоканальное устройство) с именем BUSY. Эта память имеет столько каналов, сколько в модели операторов агрегатов (в примере 6.3). Перед началом работы оператора транзакт занимает канал памяти BUSY (оператор переходит в состояние *занято*) и освобождает канал памяти BUSY в случае ожидания крана (переходит в состояние *простои*). Таким образом, нагрузка памяти BUSY соответствует среднему коэффициенту занятости

оператора. Окончательный вариант программы этого примера показан на рис. 83.

```

PROC1  FUNCTION  RN1,C5      ;Время выполнения процедуры 1
0,60/.12,70/.48,80/.83,90/1,100
PROC2  FUNCTION  RN1,C4      ;Время выполнения процедуры 2
0,80/.24,90/0.73,100/1,110
BUSY   STORAGE   7          ;Семь операторов
* Сегмент операторов агрегата доводки
  GENERATE      ,,,7,1      ;Семь операторов с приоритетом 1
  ENTER         BUSY        ;Вхождение в состояние занятости
CYCLE   ADVANCE  FN$PROC1    ;Выполнение процедуры 1
  LEAVE         BUSY        ;Вхождение в состояние простоя
  SEIZE         KRAN        ;Получение крана
  ENTER         BUSY        ;Вхождение в состояние занятости
  ADVANCE       15,5        ;Поворот отливки
  RELEASE       KRAN        ;Освобождение крана
  PRIORITY      0           ;Изменение приоритета
  ADVANCE       FN$PROC2    ;Выполнение процедуры 2
  LEAVE         BUSY        ;Вхождение в состояние простоя
  SEIZE         KRAN        ;Получение крана
  ENTER         BUSY        ;Вхождение в состояние занятости
  ADVANCE       30,5        ;Выгрузка, укладка, загрузка
  RELEASE       KRAN        ;Освобождение крана
  PRIORITY      1           ;Изменение приоритета
  TRANSFER      ,CYCLE      ;Переход к следующей отливке
* Сегмент таймера
  GENERATE      2400
  TERMINATE     1

```

Рис. 83. Окончательный вариант модели из примера 6.3

В табл. 32 показано сравнение результатов моделирования в случае использования динамического изменения приоритетов («первая операция – кратчайшая») и без него (общая очередь для всех этапов). Для получения второго варианта следует лишь удалить из модели строки с блоками PRIORITY. Количество выполненных отливок определяется по количеству транзактов, прошедших блок TRANSFER.

Таким образом, для случая использования динамических приоритетов нагрузка крана и операторов возрастает, следовательно количество готовых изделий также возрастает, т. е. система работает более эффективно.

Таблица 32. Сравнительный анализ двух способов организации очереди в модели примера 6.3

Показатели	Первая операция – кратчайшая	Общая очередь
Нагрузка крана	0,971	0,953
Средняя нагрузка операторов	0,719	0,688
Количество выполненных отливок	51	49

Задание 6.3. Из папки *Примеры* загрузите *Пример 63.gps*. Используйте эту модель, чтобы определить сколько агрегатов должно приходиться на 1 кран.

Примечание. Проведите моделирование для случая трех, четырех, пяти, шести, семи агрегатов и сравните полученные результаты.

В тетради для лабораторных работ должна быть зафиксирована сравнительная таблица результатов и записан вывод.

6.4. Выбор элементов по их состояниям

Рассмотрим систему массового обслуживания с несколькими каналами, перед которыми образуются отдельные очереди (рис. 84).

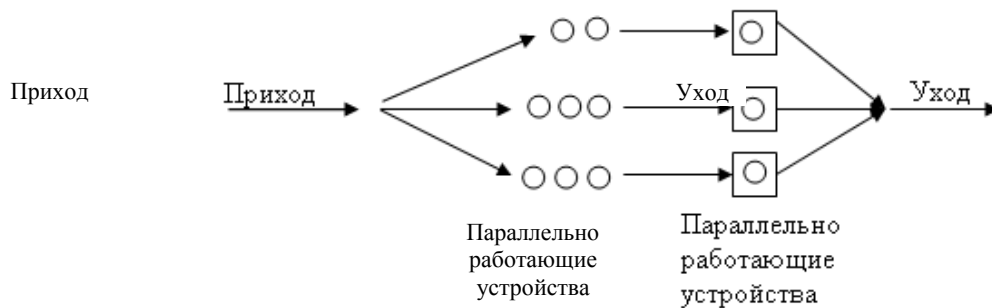


Рис. 84. СМО с несколькими очередями и несколькими параллельно работающими устройствами

Моделирование такой системы на GPSS требует использования n устройств и n очередей. Для того, чтобы избежать использования в модели n практически одинаковых веток алгоритма с последовательностью блоков QUEUE – SEIZE – DEPART – ADVANCE – RELEASE, запишем в параметр 1 активного транзакта номер выбираемой ветки (очередь плюс устройство), а затем используем этот параметр в качестве аргумента блоков QUEUE, SEIZE, DEPART и RELEASE (рис. 85).

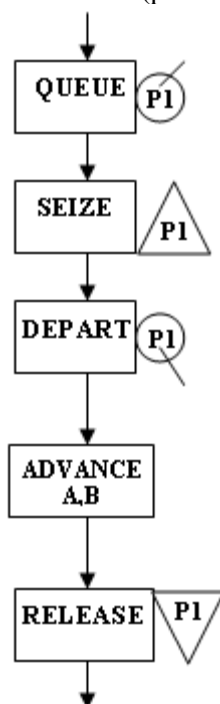


Рис. 85. Последовательность блоков моделирования нескольких устройств с отдельными очередями

Остается решить вопрос о том, каким образом выбрать число, записываемое в параметр 1 активного транзакта. Очевидно, что в простейшем случае выбор осуществляется путем анализа состояния системы:

- если одно из устройств свободно, транзакт направляется на это устройство;
- если все устройства заняты, транзакт присоединяется к наиболее короткой очереди и остается в ней до тех пор, пока не будет обслужен.

Такой выбор можно осуществить с помощью блока SELECT, который выбирает объект и помещает его номер в параметр активного транзакта. Блок SELECT имеет следующий формат:

SELECT O A,B,C,D,E,F

Значения операндов данного блока:

O – оператор условия или логический оператор; некоторые из его значений приведены в табл. 33, а остальные можно найти в справочнике системы;

A – имя (номер) параметра транзакта, которому присваивается номер выбранного объекта;

B, C – наименьший и наибольший номера объектов из просматриваемой группы;

D – значение, с которым должен сравниваться атрибут, указанный в операнде E;

E – групповое имя стандартного числового атрибута;

F – имя блока, в который передается транзакт, если ни один из членов группы не отвечает установленному условию (может быть опущен).

Таблица 33. Возможные значения операнда O блока SELECT

Оператор	Интерпретация
E	Операнд E равен операнду D?
G	Операнд E больше операнда D?
GE	Операнд E больше операнда D или равен ему?
L	Операнд E меньше операнда D?
LE	Операнд E меньше операнда D или равен ему?
NE	Операнд E не равен операнду D?
MAX	Определяется операнд E, наибольший из просмотренных (операнд D не требуется)
MIN	Определяется операнд E, наименьший из просмотренных (операнд D не требуется)

Примеры

SELECT E 1,1,3,0,F,NOFREE

Среди устройств с номерами от 1 до 3 выбирается первое, для которого значение СЧА F (состояние устройства) равно нулю, т. е. это устройство свободно. Номер выбранного устройства записывается в параметр 1 активного транзакта. Если все устройства заняты, то транзакт передается к блоку с именем NOFREE, а значение параметра 1 не изменяется.

SELECT LE LowQu,Q1,Q5,2,Q

Среди последовательно перебираемых очередей с именами от Q1 до Q5 выбирается первая очередь, для которой текущее содержимое (СЧА Q) не превышает двух, т. е. меньше или равно двум. Номер этой очереди записывается в параметр транзакта LowQu, причем если этот параметр еще не существует, то он создается. Альтернативный выход транзакта не предусмотрен: если не найдена подходящая очередь, то параметр транзакта получает значение 0 и отправляется в следующий блок. Для корректного использования этого оператора именами очередей в начале программы должны быть присвоены числовые значения с помощью операторов, например:

Q1 EQU 1
Q2 EQU 2
.....
Q5 EQU 5

SELECT MIN 2,1,4,,Q

Среди очередей с номерами от 1 до 4 выбирается очередь, для которой СЧА Q (текущее количество транзактов в очереди) является наименьшим. Номер этой очереди присваивается параметру 2 активного транзакта.

Пример 6.4. Сравнение альтернативных вариантов систем обслуживания в банке. Приход клиентов в банк является пуассоновским с интенсивностью 200 клиентов в час. В течение всего времени работы открыто 8 окошек кассиров. К каждому кассиру стоит очередь. Если в момент прихода клиента в банк хотя бы один кассир свободен, клиент сразу же попадает к этому кассиру. В противном случае клиент присоединяется к любой очереди, которая на текущий момент является кратчайшей. Обслуживание клиентов в очереди осуществляется по принципу «первым пришел – первым обслужен». После обслуживания клиент покидает банк.

Обслуживание в кассе может быть разделено на 5 различных видов. Относительная частота этих видов обслуживания и соответствующее среднее время обслуживания приведены в табл. 34. Каждый вид времени обслуживания имеет экспоненциальный закон распределения. Ни один из клиентов не требует выполнения более чем одного вида обслуживания за один визит в банк.

Таблица 34. Среднее время выполнения банковской операции

Виды операций	Среднее время обслуживания, с	Относительная частота	Суммарная частота
1	45	0,10	0,10
2	75	0,19	0,29
3	100	0,32	0,61
4	150	0,24	0,85
5	300	0,15	1,00

Директор банка заметил, что очереди в банке непомерно велики. Он хотел бы сократить время ожидания клиентов, но без привлечения к работе новых кассиров. В одном из соседних банков была введена система с «быстрой очередью». В такой системе клиенты становятся в общую очередь. Когда какой-либо кассир освобождается, клиент, стоящий в очереди первым, идет к окошку этого кассира. Директор хотел бы узнать, насколько система с быстрой очередью может сократить время ожидания клиентов в его банке.

Для решения этой задачи необходимо построить две модели: одну с общей очередью к многоканальному устройству, а вторую с отдельными очередями к кассирам. Первая модель уже была построена нами ранее. Вторая модель будет использовать последовательность блоков рис. 85 и перед ними блоки SELECT для выбора кассира или очереди.

Сложность задачи состоит в том, что для корректного сравнения систем обслуживания условия моделирования должны быть абсолютно одинаковыми. Это означает, что если, например, в первой модели 21-й клиент приходит в банк в момент времени 960 с и требует 165 с для обслуживания, то тоже самое должно быть и во второй модели. Для этого последовательность розыгрыша случайных величин в моделях должна быть одинаковой. Если использовать программу общей очереди, которая рассматривалась ранее, то время прихода нового клиента разыгрывается в момент входа транзакта в блок GENERATE, а время обслуживания – в момент входа в блок ADVANCE. При единой очереди для транзакта, который пришел 21-м, розыгрыш времени обслуживания также является 21-м. В модели с различными очередями относительный порядок, по которому клиенты попадают на обслуживание, может быть другим. Если разыгрывать время обслуживания в блоке ADVANCE, то для 21-го транзакта розыгрыш времени обслуживания не обязательно будет 21-м, а может быть, например, 18-м, если его очередь пошла быстрее. Тогда полученное время обслуживания будет отличаться от времени в первой модели.

Чтобы обеспечить одинаковые условия для обеих моделей, будем время обслуживания разыгрывать сразу же по приходу транзакта в систему, и записывать это время в параметр 1 транзакта (блок ASSIGN). В блоке же ADVANCE будем использовать значение этого параметра для определения времени обслуживания.

Программа модели с общей очередью приведена на рис. 86. За единицу времени принята 0,1 с. Средний интервал прихода транзактов в систему равен $60 \cdot 60 \cdot 10 : 200 = 180$ ед. Функция Mean служит для розыгрыша среднего времени обслуживания транзакта, согласно табл. 34 (с учетом единицы измерения времени). Многоканальное устройство (память) имеет имя Kassy, а единая очередь называется One.

```
Mean FUNCTION RN1,D5 ;Ф-я сред. вр. обслуж.
.1,450/.29,750/.61,1000/.85,1500/1,3000
Kassy STORAGE 8 ;Восемь касс обслуж.
* 1-й сегмент - клиенты
GENERATE (Exponential(1,0,180)) ;Приход клиентов
ASSIGN 1, (Exponential(1,0,FN$Mean));Парам. 1←вр. обл.
QUEUE One ;Регистр. в един. очер.
ENTER Kassy ;Занятие кассира
DEPART One ;Выход из очереди
ADVANCE P1 ;Задер. на вр. из парам. 1
LEAVE Kassy ;Освобождение кассира
TERMINATE ;Уход из банка
* 2-й сегмент - таймер
GENERATE 180000 ;Таймер приход. через 5 ч
TERMINATE 1 ;Прекращ. моделирования
```

Рис. 86. Модель обслуживания клиентов в «быстрой очереди»

В модели с отдельными очередями после прихода транзакта и розыгрыша времени его обслуживания решается вопрос о том, в какую очередь направить этот транзакт. Для этого используются блоки SELECT:

```
SELECT E 2,1,8,0,F,Queue
```

Среди устройств с номерами от 1 до 8 выбирается первое свободное устройство ($F = 0$) и его номер записывается в параметр 2 активного транзакта. Если свободных устройств нет, то транзакт передается на блок с именем Queue.

```
Queue SELECT MIN 2,1,8,,Q
```

Блок с именем Queue проверяет очереди с номерами от 1 до 8 и выбирает среди них очередь с наименьшим текущим содержимым (СЧА Q). Номер этой очереди записывается в параметр 2 активного транзакта.

Далее значение параметра 2 используется в блоках QUEUE, SEIZE, DEPART и RELEASE. Кроме восьми отдельных очередей к каждому устройству с номерами от 1 до 8 в модели имеется общая очередь с номером 10, которая собирает агрегированную статистику по клиентам, пришедшим в банк. Программа системы с отдельными очередями к кассам приведена на рис. 87.

```
Mean FUNCTION RN1,D5 ;Ф-я опред. сред. вр. обслуж.
.1,450/.29,750/.61,1000/.85,1500/1,3000
* 1-й сегмент - клиенты
GENERATE (Exponential(1,0,180)) ;Приход клиентов
ASSIGN 1, (Exponential(1,0,FN$Mean));Парам. 1←вр. обл.
SELECT E 2,1,8,0,F,Queue ;Есть ли свободный кассир?
Line QUEUE P2 ;Зарегист. в очер. к кассиру
```

QUEUE	10	;Зарегистрир. в общ. очереди
SEIZE	P2	;Занять кассира
DEPART	10	;Покинуть общ. очередь
DEPART	P2	;Покинуть очередь к кассиру
ADVANCE	P1	;Задержка на вр. из парам. 1
RELEASE	P2	;Освобождение кассира
TERMINATE		;Уход из банка
QueueP	SELECT MIN 2,1,8,,Q	;Парам. 2←ном. сам. малой оч.
	TRANSFER ,Line	;Переход на обслуживание
* 2-й сегмент – таймер		
	GENERATE 180000	;Таймер приходит через 5 ч
	TERMINATE 1	;Завершение прогона

Рис. 87. Модель обслуживания клиентов в отдельных очередях

Для каждой модели было выполнено по 5 прогонов (пять 5-часовых рабочих дней), между которыми использовалась команда CLEAR. Эта команда сбрасывает до нуля накопленную статистику моделирования и удаляет транзакты из цепей будущих и текущих событий. Таким образом, при отсутствии транзактов моделирование каждый раз начиналось сначала. Однако команда CLEAR не инициализирует генераторы случайных чисел, поэтому в каждом прогоне получается разный набор разыгранных случайных величин, характеризующий различную нагрузку на систему в каждый из дней. Результаты моделирования приведены в табл. 35.

Таблица 35. Средняя длина очереди для двух методов организации очередей в банке

Время моделирования	Быстрая очередь, с	Отдельные очереди в кассы, с
1-й день	75,236	129,628
2-й день	51,923	87,613
3-й день	225,752	282,895
4-й день	812,920	862,064
5-й день	439,202	451,590
Математическое ожидание	321,007	362,758

Анализ этой таблицы показывает, что для всех случаев моделирования система быстрой очереди обеспечивает меньшее среднее время ожидания. В среднем сокращение времени ожидания составляет 12%. Кроме того, из таблицы ясно, насколько важно обеспечивать одинаковые условия моделирования. Предположим, что мы получили результат, который соответствует второму дню системы с несколькими очередями. А для системы с одной очередью взяли результат четвертого дня. Тогда можно сделать ошибочный вывод о преимуществе системы с несколькими очередями.

Задание 6.4. Из папки *Примеры* загрузите файлы *Пример641.gps* и *Пример642.gps*. Самостоятельно получите результаты за первые 3 дня и сравните их со значениями в табл. 35.

6.5. Резидентное и транзитное время транзактов

Резидентным временем транзакта называется интервал времени, в течение которого транзакт находится в модели. При входе транзакта в модель фиксируется время этого входа, значение которого называется *отметкой времени*. Значение этой отметки недоступно для программиста, однако он может использовать стандартный числовой атрибут M1, который равен текущему значению таймера модельного времени минус отметка времени. Таким образом, значение СЧА M1 и является резидентным временем транзакта.

Транзитным временем транзакта называется интервал времени, в течение которого транзакт переходит от одной произвольно выбранной точки модели до другой. Допустим, необходимо определить интервал времени, в течение которого транзакт проходит от точки А до точки В. Для этого следует выполнить следующие действия:

- Когда транзакт находится в точке А модели, нужно сохранить в параметре транзакта значение таймера модельного времени. Для этого используется блок MARK (рис. 88), имеющий формат

MARK A

Операнд А – номер параметра, в который записывается значение таймера модельного времени.

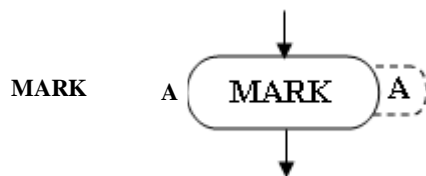


Рис. 88. Графическое изображение блока MARK (Отметить)

- Когда транзакт достигает точки В, рассчитывается время, которое прошло с момента отметки транзакта блоком MARK. Стандартный числовой атрибут MP_j (где j – номер параметра, в котором было записано время блоком MARK) дает разницу между текущим значением таймера модельного времени и временем, записанным в параметре j . Таким образом, СЧА MP_j дает значение транзитного времени транзакта между точками А и В.

В случае, когда в блоке MARK параметр А опущен, отметка времени транзакта становится равна текущему модельному времени.

6.6. Использование таблиц в языке GPSS

Объект *таблица* позволяет получать эмпирические таблицы распределения исследуемых случайных величин, а также некоторые количественные характеристики этих распределений (математическое ожидание и среднееквадратическое отклонение).

Предположим, исследуется время прохождения транзакта между точками А и В модели. В процессе моделирования многие транзакты совершают этот переход. Если записать транзитное время каждого транзакта в процессе моделирования, то получится некоторое множество наблюдаемых значений случайной величины *Время перехода транзакта из А в В*. Такое множество значений называется *выборкой*. Для выборки рассчитывается среднее значение (оценка математического ожидания), стандартное отклонение (оценка среднееквадратического отклонения), а также частоты попадания значений в некоторые заданные интервалы. По этим частотам в дальнейшем может быть построена гистограмма значений выборки.

Оператор описания TABLE позволяет объявить таблицу для сбора значений по некоторой выборке случайной величины:

NAME TABLE A,B,C,D

Операнды оператора имеют следующие значения:

NAME – символическое или числовое имя таблицы;

A – аргумент таблицы (имя переменной или СЧА), который должен учитываться в таблице;

B – верхняя граница первого интервала;

C – ширина всех промежуточных интервалов;

D – общее число интервалов таблицы, включая левый и правый.

На рис. 89 показано, как разбивается вся числовая ось на интервалы согласно оператору TABLE.

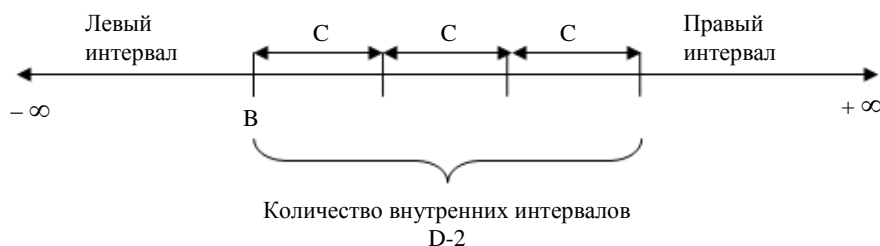


Рис. 89. Разделение числовой оси на ряд интервалов

Оператор описания таблицы TABLE должен располагаться в начале программы.

Занесение информации в таблицу происходит тогда, когда транзакт входит в блок TABULATE (рис. 90), который имеет формат

TABULATE A,B

Операнды блока имеют следующие значения:

A – имя таблицы, символическое или числовое;

B – вес, т. е. число добавляемых в соответствующий интервал единиц (по умолчанию 1).

В операторе TABULATE не указывается, какая именно величина заносится в таблицу, так как это уже задано в операторе TABLE.

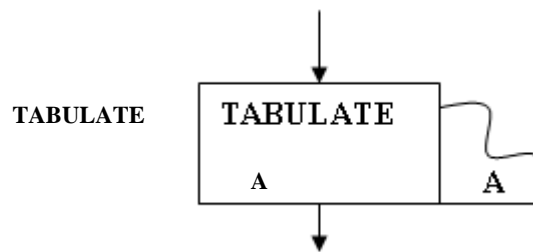


Рис. 90. Графическое изображение блока TABULATE

Примеры

```
CORE TABLE $$KANAL,0,50,10
```

Таблица CORE предназначена для получения частотного распределения текущего содержимого памяти KANAL, причем числовая ось разбивается на 10 интервалов: $(-\infty, 0]$, $(0, 50]$, $(50, 100]$, ..., $(400, +\infty)$.

```
TABULATE CORE,2
```

Когда транзакт проходит этот блок, в соответствующий интервал таблицы CORE добавляется 2.

Таблицу для табулирования времени пребывания в очереди можно описать с помощью оператора QTABLE. Он имеет формат, аналогичный формату оператора TABLE, только операнд A содержит имя (номер) наблюдаемой очереди. В этом случае можно обойтись без оператора TABULATE, его функции будет выполнять оператор DEPART.

Пример

```
QTIME QTABLE TIME,100,100,10
```

```
.....
QUEUE TIME
```

```
.....
DEPART TIME
```

При входе транзакта в блок DEPART время пребывания в очереди TIME автоматически заносится в таблицу QTIME.

По окончании моделирования собранная информация по таблице распечатывается в отчете.

С таблицами связаны следующие числовые атрибуты:

TV_j или TV\$имя – среднее значение элементов таблицы;

TC_j или TC\$имя – число учтенных в таблице элементов;

TD_j или TD\$имя – стандартное отклонение элементов таблицы.

Пример 6.5. Обратимся вновь к примеру 6.2 (модели продовольственного магазина). Допустим, что нужно оценить время пребывания покупателей в магазине. Для этого к модели *Пример 62.gps* добавьте следующее:

- В начале программы, до начала первого сегмента, объявите таблицу с именем RTIME, которая будет фиксировать резидентное время пребывания транзакта в модели (M1). Эта таблица рассчитана на 9 интервалов, верхняя граница левого интервала и ширина промежуточных интервалов равны 100:

```
RTIME TABLE M1,100,100,9
```

- Перед оператором TERMINATE в сегменте покупателей поставьте оператор, который будет регистрировать информацию по проходящему транзакту в таблице:

```
TABULATE RTIME
```

Выполните компиляцию и прогон модели. Изучите информацию по таблице RTIME в стандартном отчете (рис. 91).

Пример 55.12.1 - REPORT							
TABLE	MEAN	STD.DEV.	RANGE		RETRY FREQUENCY	CUM. %	
RTIME	310.703	119.664			0		
			100.000	100.000	12	3.51	
			200.000	200.000	60	21.05	
			300.000	300.000	86	46.20	
			400.000	400.000	95	73.98	
			500.000	500.000	72	95.03	
			600.000	600.000	17	100.00	

Рис. 91. Информация по таблице в отчете

В столбцах отчета указывается следующая информация:

- TABLE – имя таблицы;
- MEAN – среднее значение (оценка математического ожидания);

- **STD.DEV.** – стандартное отклонение по выборке (оценка среднеквадратического отклонения случайной величины);
- **RANGE** – интервалы таблицы;
- **RETRY** – количество транзактов, ожидающих наступления особых условий, связанных с состоянием таблицы;
- **FREQUENCY** – частота попадания исследуемой величины в соответствующий интервал;
- **CUM.%** – суммарный процент попадания величины в интервал от минус бесконечности до верхней границы соответствующего интервала.

Если в последние интервалы, предусмотренные оператором **TABLE**, не попадает ни одного значения, то они не выводятся в отчете. Так, в нашем примере предусмотрено 9 интервалов, а выведено только 6.

За состоянием таблицы можно наблюдать в динамике моделирования с помощью окна *Таблиц*, которое вызывается командой **Window/Simulation Window/Table Window**. Далее в диалоговом окне из списка нужно выбрать таблицу, гистограмму которой нужно вывести на экран. Кроме гистограммы, в окне *Таблиц* выводятся значения среднего и стандартного отклонения по выборке (рис. 92).

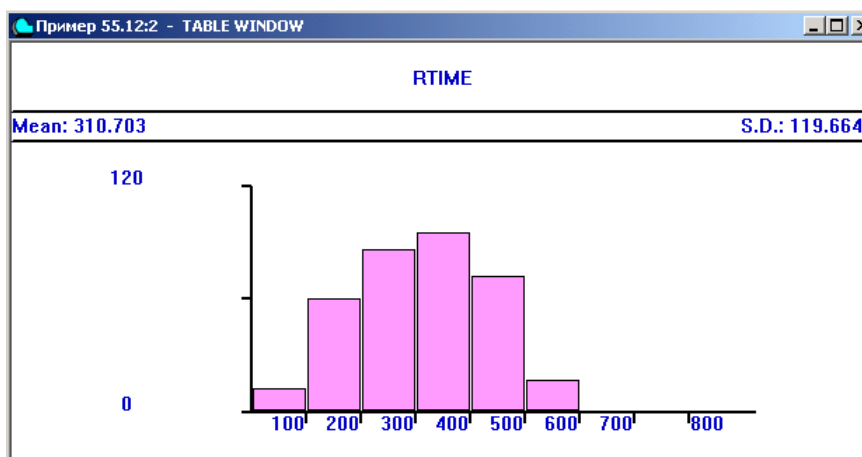


Рис. 92. Окно таблицы **RTIME**

Задание 6.5. Измените модель производства деталей с обжигом в печи (*Пример 44.gps*) таким образом, чтобы с ее помощью можно было оценить распределение для следующих случайных величин:

1. Время сборки детали.
2. Время обработки детали (состоит из двух компонентов: времени сборки и времени обжига).
3. Полное время обработки детали (состоит из трех компонентов: времени сборки, времени ожидания освобождения печи и времени обжига).

Выполните моделирование для случая четырех сборщиков. Укажите, насколько хорошо согласуется распределение времени сборки детали с исходными данными задачи (равномерное распределение 30 ± 5). Используя среднее значение полного времени обработки детали, рассчитайте число деталей, сборка которых должна закончиться в течение периода моделирования. Уточните, согласуется ли это число со значениями счетчиков блоков.

В тетради для лабораторных работ приведите измененный текст программы модели, гистограммы таблиц для случайных величин 1–3 данного задания, а также ответы на вопросы.

6.7. Задания для самостоятельной работы

Вариант 1

Интервалы прибытия кораблей в порт, работающий круглосуточно 7 дней в неделю, показаны в табл. 36.

Таблица 36. Распределение интервалов прихода кораблей в порт

Интервалы прихода, мин	Суммарная частота
Менее 125	0,00
175	0,10
225	0,22
275	0,40
325	0,55
375	0,78
425	0,90

Тридцать процентов этих кораблей оснащены современным оборудованием, позволяющим производить быструю погрузку (выгрузку). Эти корабли причаливают к первому перегрузочному комплексу (рис. 93) с двумя причалами, на каждом из которых перегрузка может быть закончена за $(1,0 \pm 0,1)$ сут. Остальные корабли направляются ко второму перегрузочному комплексу, у которого есть 6 причалов. На них перегрузка длится $(1,5 \pm 0,2)$ сут. Для ввода кораблей в порт используют одно буксировочное судно. Время буксировки к причалу составляет (30 ± 10) мин, а от причала (15 ± 5) мин. Необходимо промоделировать работу такой системы в течение 100 сут. Требуется оценить следующее:

- эмпирическую функцию распределения общего времени ожидания у входа в порт кораблей, оснащенных современным оборудованием (это время состоит из ожидания освобождения причала и ожидания буксировки);
- эмпирическую функцию распределения общего времени ожидания у входа в порт всех прочих кораблей;
- степень использования причалов первого и второго комплексов, а также буксировочного судна.

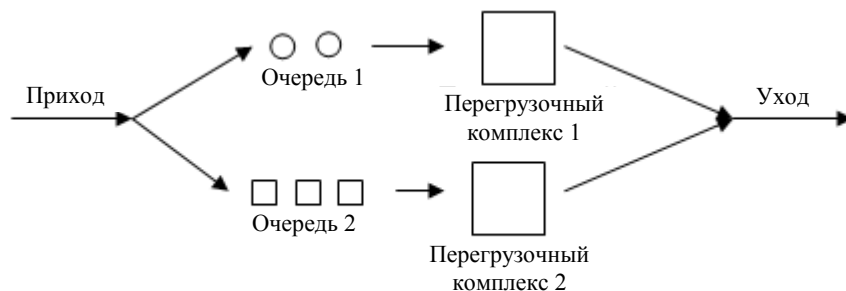


Рис. 93. Структура моделируемой системы

Вариант 2

В парикмахерскую с одним парикмахером клиенты приходят с интервалом времени, который подчиняется нормальному закону распределения, со средним значением 12 мин и дисперсией 3. Время обслуживания клиентов распределено экспоненциально. Однако парикмахер устает в течение дня и среднее время обслуживания клиентов непрерывно увеличивается: вначале обслуживания оно равно 11 мин, к концу второго часа составляет 14 мин, к концу четвертого часа увеличивается до 17 мин, к концу седьмого часа достигает 25 мин, а к моменту окончания работы (конец восьмого часа) будет равно 30 мин. Владелец парикмахерской предполагает, что если дать парикмахеру получасовой перерыв после четырех часов обслуживания клиентов (как только он закончит обслуживание очередного клиента), то результаты работы будут лучше. Дело в том, что после перерыва парикмахер с новыми силами вступает в работу и может обслуживать клиентов так, как и с начала рабочего дня. При этом предполагается, что клиенты ждут окончания перерыва. Выполните моделирование работы парикмахерской в течение 8-часового рабочего дня для двух случаев: без обеденного перерыва и при наличии перерыва. Сравните результаты и сделайте вывод.

Вариант 3

На таможенном пункте находится 7 касс. Каждая из них может обслужить автомобиль за (15 ± 3) с. Предположим, что в 15 ч дня открыты 4 кассы и ни один автомобиль не стоит в очереди. Поток автомобилей, прибывающих на таможенный пункт, имеет экспоненциальный закон распределения. Средняя интенсивность этого потока к концу рабочего дня возрастает, а вечером падает (табл. 37). При увеличении потока автомобилей для уменьшения простоев в 16 ч 30 мин открывают пятую кассу, а в 17 ч – оставшиеся две кассы.

К каждой кассе существует отдельная очередь. Автомобиль, прибывающий на таможенный пункт, выбирает свободную кассу, если она есть. В противном случае он присоединяется к самой короткой очереди. Выполните моделирование работы таможенного пункта в течение рабочего дня и определите максимальное и среднее число автомобилей в каждой очереди, а также соответствующие показатели по системе в целом.

Таблица 37. Изменение потока автомобилей во времени

Период, ч.мин	Интенсивность потока, автомобилей в час
15.00–16.00	1 000
16.00–16.30	1 250
16.30–17.00	1 600
17.00–17.30	2 000

17.30–18.00	1 800
18.00–18.30	1 300
18.30–19.00	1 000

Вариант 4

На обрабатывающий участок цеха детали поступают в среднем через каждые 50 мин. Первичная обработка деталей производится на одном из двух станков. Первый станок обрабатывает деталь в среднем 40 мин и имеет 4% брака, а второй соответственно 60 мин и 8% брака. Все интервалы времени распределены по экспоненциальному закону. Бракованные детали возвращаются на повторную обработку на второй станок, при этом они обслуживаются вне очереди и вероятность брака снижается до 6%. Детали, попавшие в разряд бракованных дважды, считаются отходами.

Выполните моделирование обработки 1 000 деталей. Определите эмпирическую функцию распределения времени обработки деталей, исключая отходы, вероятность появления отходов и загрузку каждого станка.

Вариант 5

В морском порту есть 2 причала: старый и новый. У старого причала могут швартоваться 2 судна. Здесь работают 2 порталных крана, производящие разгрузку судна со скоростью 5 т/ч. У нового причала имеется место для 5 судов. Здесь работают 3 крана, производящие разгрузку со скоростью 10 т/ч. Судна прибывают в акваторию порта со средним интервалом времени 5 ч, причем интервал их прихода имеет экспоненциальный закон распределения. Прибывающее судно может иметь емкость 100, 200 или 300 т с вероятностями 0,3; 0,5 и 0,2 соответственно.

В ожидании места у причала судно бросает якорь на рейде. Для швартовки и отхода судна от причала требуется по 1 ч времени. Судно, попавшее на новый причал, может ожидать освобождения крана. Таким образом, общее время ожидания судна складывается из ожидания освобождения причала и ожидания освобождения крана, если оно было. Постройте эмпирические функции распределения для времени ожидания места у причала, времени ожидания крана и общего времени ожидания судов. Выполните моделирование до того момента, когда 200 кораблей покинут порт.

С помощью модели определите, какое решение лучше: добавить еще один кран на новом причале либо модернизировать краны на старом причале, чтобы они выполняли разгрузку со скоростью 7 т/ч.

Вариант 6

В справочной телефонной сети города работают 5 операторов, сообщая номера телефонов по запросам абонентов, которые обращаются по одному номеру 09. Автоматический коммутатор переключает абонента на того оператора, в очереди которого ожидает наименьшее количество абонентов, причем наибольшая допустимая длина очереди перед оператором – 2 абонента. Если все очереди имеют максимальную длину, вновь поступивший вызов получает отказ. Обслуживание абонентов операторами длится (30 ± 20) с.

Для определения интервала поступления вызовов были проведены дополнительные исследования, результаты которых показаны в табл. 38. Известно также, что функция распределения интервала между вызовами является непрерывной.

Таблица 38. Результаты изучения потока вызовов на обслуживание

Интервал между вызовами, с	Частота попадания
Менее 2	0,0
2÷4	0,2
4÷6	0,5
6÷8	0,3

Промоделируйте обслуживание 200 вызовов. Подсчитайте количество отказов. Определите коэффициенты загрузки операторов справочной.

Вариант 7

Диспетчер управляет внутризаводским транспортом и имеет в своем распоряжении 2 грузовика. Заявки на перевозки поступают к диспетчеру каждые (5 ± 4) мин. С вероятностью 0,5 диспетчер запрашивает по радио один из грузовиков и передает ему заявку, если тот свободен. В противном случае он запрашивает другой грузовик и таким образом продолжает сеансы связи, пока один из грузовиков не освободится. Каждый сеанс связи длится ровно минуту. Диспетчер допускает накопление у себя до 5 заявок, после чего вновь прибывшие заявки получают отказ. Время выполнения заявки грузовиком имеет экспоненциальный закон распределения, причем среднее значение этого времени зависит от типа заявки. Всего возможно появление заявок трех типов, информация по ним приведена в табл. 39.

Таблица 39. Данные по заявкам на транспортные перевозки

Типы заявок	Вероятность появления заявки данного типа	Среднее время обслуживания заявки, мин
1-й	0,4	8
2-й	0,4	10
3-й	0,2	16

Выполните моделирование работы внутризаводского транспорта в течение 10 ч. Подсчитайте число заявок, обслуженных каждым грузовиком, и количество отклоненных заявок. Определите коэффициенты загрузки грузовиков.

Вариант 8

Небольшое кафе на 10 столиков по 4 человека стоит на «бойком» месте. В среднем каждые 15 мин сюда приходят компании клиентов. Интервал времени между приходами имеет экспоненциальный закон распределения. Компания может состоять из одного, двух, трех или четырех человек с вероятностями, показанными во втором столбце табл. 40. Столик выделяется для компании, и администрация не имеет право подсаживать клиентов из другой компании за уже занятый столик. Время, которое компания проводит в кафе, распределено экспоненциально. Однако при этом среднее время обеда (ужина) зависит от размера компании, как показано в третьем столбце табл. 40.

Если все столики в кафе заняты, то 30% пришедших компаний согласны подождать в баре от 20 до 40 мин, это время имеет равномерное распределение. После этого они делают еще одну попытку занять столик в кафе. Если и эта попытка неудачна, компания уходит искать другое кафе.

Кафе открыто с 11 ч до 23 ч. Выполните моделирование работы кафе в течение одного рабочего дня и рассчитайте среднюю дневную выручку, если известно, что один клиент приносит прибыль в среднем 10 тыс. р. за час его пребывания в кафе.

Примечание. Ориентируйтесь на среднее количество клиентов в кафе в течение дня.

Определите, какое количество компаний не попало в кафе, и для скольких из них вторая попытка занять столик оказалась успешной.

Таблица 40. Характеристики компаний, приходящих в кафе

Количество человек в компании	Вероятность прихода	Среднее время обслуживания, мин
1	0,2	40
2	0,5	90
3	0,2	180
4	0,1	300

Контрольные вопросы

1. Можно ли в GPSS использовать функцию от случайного аргумента и от детерминированного?
2. Нарисуйте график следующей функции и объясните его смысл:
MYWAIT FUNCTION Q\$WAIT,D4
1,10/3,7/6,4/15,1
3. Нарисуйте график функции
MYWAIT FUNCTION Q\$WAIT,C4
1,10/3,7/6,4/15,1
4. Какое количество параметров может иметь транзакт? Как можно обратиться к параметру?
5. Какой блок увеличивает значение параметра с номером 3 на 10? Напишите его.
6. Для чего в языке GPSS служат переменные? Как объявить и использовать переменную?
7. Как определить уровень приоритета активного транзакта? Можно ли изменить приоритет транзакта в ходе моделирования? Если можно, то как это сделать?
8. Поясните работу оператора
SELECT E 1,1,3,0,F,NOFREE
9. Поясните работу оператора
SELECT MIN 2,1,8,,Q
10. Что такое резидентное и транзитное времена транзакта и как они определяются?
11. Для чего предназначен объект *таблица* в языке GPSS? Какая информация по таблице распечатывается в отчете?
12. Поясните работу оператора
TABMAST TABLE FC\$Master,0,5,10
Как заносится информация в такую таблицу?

15. Как можно наблюдать за состоянием таблицы в динамике моделирования?

ОРГАНИЗАЦИЯ ИМИТАЦИОННЫХ ЭКСПЕРИМЕНТОВ

7.1. Переходный период стохастического процесса

Наиболее распространенная ошибка при имитационном моделировании – недостаточное внимание, уделяемое оценке выходных данных. Много усилий тратится на программирование модели, затем выполняется один прогон и полученные оценки рассматриваются как «истинные». Для продвижения модельного времени используются случайные выборки, поэтому полученные оценки являются лишь отдельными реализациями случайных величин, которые, возможно, имеют большую дисперсию. На самом деле, чтобы получить правильные, значимые результаты при имитационном моделировании, нужно использовать статистические методы для разработки и анализа моделирующих экспериментов.

Пусть величины Y_1, Y_2, \dots представляют стохастический (случайный) процесс, развивающийся во времени. Например, Y_j – это средняя длина очереди в течение j -го часа либо коэффициент использования устройства в течение j -х суток и т. п. Продолжительность одного прогона на модели равна m . Тогда в результате одного i -го прогона получаем реализацию этих случайных величин $y_{i1}, y_{i2}, \dots, y_{im}$. Допустим, что выполняется n прогонов, каждый с разной последовательностью случайных чисел. Предполагается, что перед началом каждого прогона статистические счетчики переводились в исходное состояние, использовались одни и те же начальные условия, а генераторы случайных чисел выдавали разные последовательности величин. Тогда получаем матрицу реализаций $[y_{ij}]$, $i = 1, \dots, n$; $j = 1, \dots, m$ (рис. 94).

$$\begin{array}{c} y_{11}, \dots, y_{1i}, \dots, y_{1m} \\ y_{21}, \dots, y_{2i}, \dots, y_{2m} \\ \vdots \\ y_{n1}, \dots, y_{ni}, \dots, y_{nm} \end{array}$$

Рис. 94. Матрица реализаций стохастического процесса

Данные наблюдений, полученные в результате определенного прогона (строка матрицы), не являются независимыми и одинаково распределенными, поэтому к ним нельзя применять классические процедуры статистического анализа. Однако данные наблюдений из j -го столбца являются независимыми и одинаково распределенными, поэтому к ним применяются обычные статистические методы.

Поведение стохастического процесса можно разделить на переходный и установившийся (стационарный) периоды. Поведение в переходный период зависит от заданных начальных условий. Например, если моделируется работа универсального магазина, то процесс изменения длины очереди в кассу зависит от числа покупателей, которые изначально находятся в торговом зале. Длительность самого переходного периода также зависит от начальных условий. Функции плотности распределения для каждого столбца матрицы $[y_{ij}]$ являются различными (рис. 95).

Поведение стохастического процесса в стационарном периоде не зависит от начальных условий, а функции распределения по столбцам матрицы реализаций являются приблизительно одинаковыми (или стремятся к одному пределу).

Задача исследования может ставиться как для стационарного, так и для переходного периода. Стационарный режим обычно интересует исследователя при разработке новой системы либо модификации старой, когда нужно оценить ее работу в течение длительного периода нормальной работы. Например, компания планирует построить новую производственную систему и хотела бы определить ее среднюю производительность за час в течение длительного периода работы.

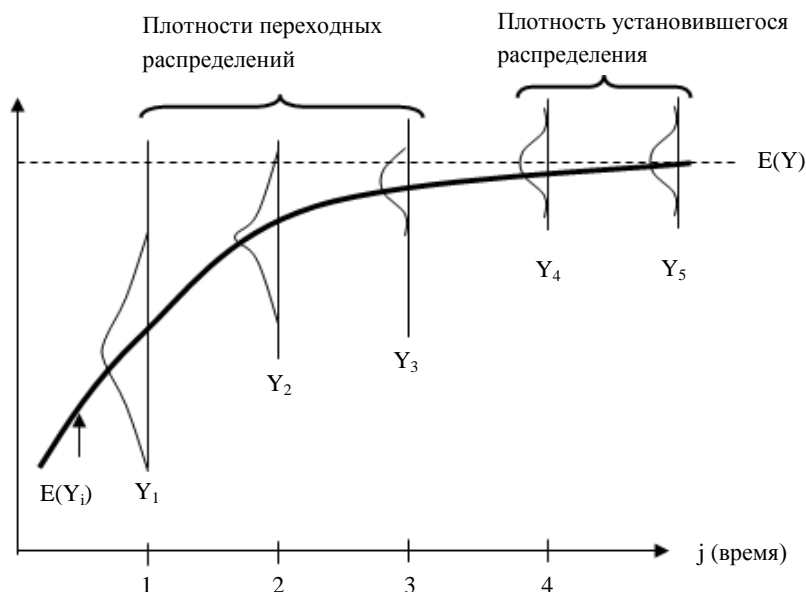


Рис. 95. Функции плотности распределения для переходного и стационарного периодов случайного процесса

Однако очень часто система не успевает войти в стационарный режим, либо он просто не существует. Например, нужно промоделировать работу банка в течение рабочего дня, причем характеристики потока клиентов изменяются в течение этого дня, имеются обеденные перерывы и т. д.

В случае, когда ставится задача исследования стационарного режима, возможны три подхода:

1. Разработать модель так, чтобы условия функционирования были типичными с самого начала. Этот подход требует от разработчика знания типичных условий работы и умения внести в модель эти условия, что не всегда реально.
2. Проводить моделирование так долго, чтобы данные, собранные вначале, «утонули» среди типичных данных и не оказали заметного влияния. Этот подход нежелателен, поскольку долгое моделирование требует значительных финансовых затрат.
3. Отбросить первые l наблюдений, пока система не войдет в стационарный режим, и не учитывать их при расчете статистических характеристик. Этот способ решения задач наиболее распространен, однако возникает вопрос, как же определить длительность переходного периода, т. е. значение l . В GPSS World можно построить график некоторой характеристики и по виду этого графика определить момент входа системы в стационарный режим.

Пример 7.1. Исследование задачи о трикотажной фабрике. В примере 4.5 рассматривается модель трикотажной фабрики, которая имеет 50 собственных машин и несколько арендуемых. Машины работают некоторое время, а затем ломаются и поступают в ремонтную мастерскую. В мастерской работают несколько рабочих, которые параллельно ремонтируют машины, беря их из общей очереди. Отремонтированные машины поступают в резерв, и в случае поломки ими заменяются нерабочие машины (см. рис. 59). Необходимо определить коэффициент загрузки многоканального устройства Nowon, которое изображает машины, находящиеся в работе.

Согласно программе, моделирование начинается в момент, когда все машины исправны и будут работать до отказа (157 ± 25) ч. Очевидно, что эти условия нетипичны, и система должна иметь переходный период. Для определения длительности этого переходного периода можно построить график СЧА $SR\$Nowon$, который показывает коэффициент использования памяти, выраженный в долях от 0 до 1 000. На рис. 96 показано, как следует заполнять диалоговое окно, появляющееся по команде **Window/Simulation Window/Plot Window...** (см. лабораторную работу 3). При этом в сегменте таймера модели установлено предельное модельное время, равное 3 тыс. ед. На рис. 97 представлен график, который будет получен после одного прогона. По этому графику можно определить, что система войдет в стационарный режим приблизительно в момент 1,5 тыс. ед., когда показатель окончательно стабилизируется.

Если в сегменте таймера модели поставить предельное время 1,5 тыс. ед., то выполнение команды

START 1,NP

позволит промоделировать работу системы до момента вхождения в стационарный период. Затем процесс моделирования будет остановлен, но отчет не сформируется (используется аргумент NP).

Сбросить статистику моделирования можно командой RESET. При этом транзакты, которые находятся в цепях текущих и будущих событий, остаются на месте, а статистические счетчики обнуляются.

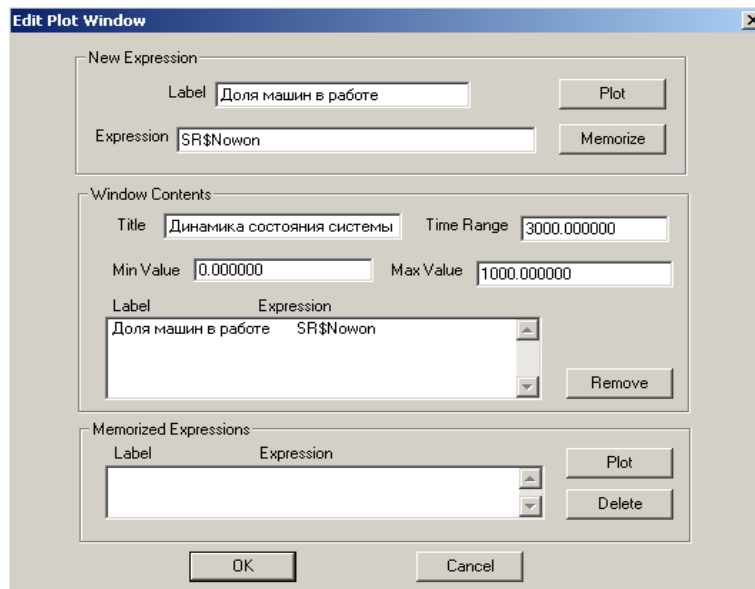


Рис. 96. Окно редактирования графика

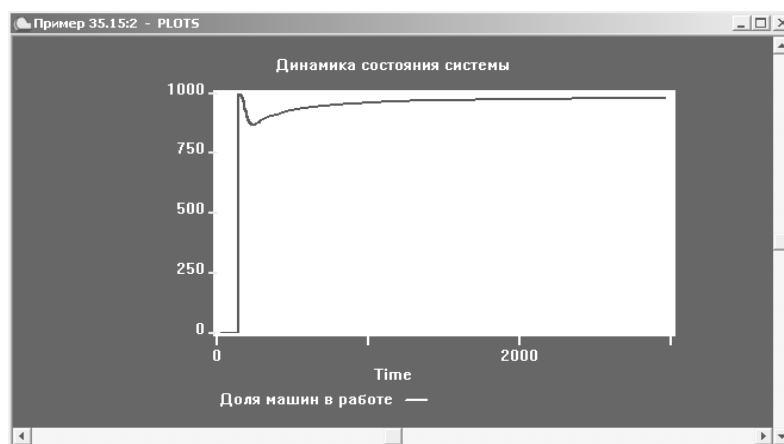


Рис. 97. График изменения доли используемых (исправных) машин

Далее для сбора статистики моделирование нужно продолжить уже по стационарному периоду. Предположим, что достаточно рассмотреть 3 тыс. ед. модельного времени стационарного периода (вопрос определения длительности моделирования в данном пособии не обсуждается). Для этого достаточно выполнить команду

START 2

После выполнения программы модели сформируется отчет, из которого можно будет взять интересующий нас показатель – коэффициент использования памяти Nowon.

Введем понятие реплики эксперимента. Будем называть *репликой* последовательность действий, которая дает возможность получить обоснованные характеристики исследуемой системы, например, без учета данных переходного периода.

Если на данной модели проводится несколько реплик, то перед началом каждой из них, кроме первой, следует очистить модель командой CLEAR – удаляются транзакты, вошедшие ранее в систему и обнуляется статистика моделирования. А затем нужно изменить начальные значения генераторов равномерных случайных чисел, чтобы последовательность генерируемых случайных величин в каждой реплике была разной. Эту задачу выполняет команда RMULT, например:

RMULT 7

В модели используется генератор RN1. Число 7 выбрано для примера, в качестве начального значения генератора RN1 можно использовать любое число. Необходимо только следить за тем, чтобы для всех реплик эти числа были разными.

Таким образом, одна реплика на данной модели будет состоять из следующей последовательности действий:

```
CLEAR
RMULT 7
START 1,NP
RESET
START 2
```

7.2. Статистическая обработка результатов моделирования

Для получения статистических оценок по стационарному периоду используются наблюдения, не входящие в переходный период работы l в ходе каждого эксперимента на имитационной модели. Допустим, было выполнено n реплик эксперимента по схеме, представленной в разделе 7.1. Для каждой реплики было получено значение некоторого показателя (например, коэффициент использования памяти Nowon). Обозначим это значение x_i , $i = \overline{1, n}$.

Эти величины являются независимыми и одинаково распределенными. Для них можно найти точечную оценку математического ожидания по формуле

$$E(x_i) \approx \bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \quad (10)$$

а также оценку дисперсии:

$$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}. \quad (11)$$

Задавшись некоторым уровнем значимости α , можно определить доверительный интервал $[\bar{x} - \beta; \bar{x} + \beta]$, в который попадает математическое ожидание величины X с вероятностью $1 - \alpha$.

Уровнем значимости называется допускаемая нами вероятность непадения математического ожидания в доверительный интервал. Обычно используется уровень значимости $\alpha = 0,05$ или $\alpha = 0,1$.

Надежностью интервальной оценки, или доверительной вероятностью $p = 1 - \alpha$, называется вероятность того, что действительное значение математического ожидания попадает в доверительный интервал.

Точностью называется половина длины доверительного интервала β . Точность доверительного интервала для математического ожидания при уровне значимости α и числе экспериментов n определяется по формуле

$$\beta = t_{\alpha, n-1} \sqrt{\frac{S^2}{n}}, \quad (12)$$

где $t_{\alpha, n-1}$ – критическое значение распределения Стьюдента со степенью свободы $n - 1$ и уровнем значимости α , которое может быть вычислено по таблице либо с помощью функции Excel СТЬЮДРАСПОБР().

Пример 7.2. Обработка результатов экспериментов на модели трикотажной фабрики. Пусть для задачи из примера 4.5. было проведено 7 реплик по схеме, описанной в разделе 7.1. Для каждой реплики получен коэффициент использования памяти Nowon. Эти результаты занесены в таблицу Excel (рис. 98).

	А	В	С
1	№ прогона	Начальное значение генератора RN1	К-т использования памяти Nowon
2	1	1	0,988
3	2	2	0,991
4	3	3	0,988
5	4	4	0,99
6	5	5	0,99
7	6	1000	0,992
8	7	2340	0,992
9			
10	Оценка матожидания		0,990142857
11	Оценка дисперсии		2,81E-06
12	Точность доверительного интервала		0,001231
13	нижняя граница		0,988912
14	верхняя граница		0,991374

СРЗНАЧ(С2:С8)

ДИСП(С2:С8)

СТЬЮДРАСПОБР(0,1;6)*КОРЕНЬ(С11/7)

Рис. 98. Статистическая обработка результатов по модели трикотажной фабрики

Для расчета статистических характеристик исследуемого показателя использовались стандартные функции Excel, которые на рис. 98 показаны возле соответствующих ячеек и соответствуют формулам (10) – (12). Число степеней свободы при семи репликах равно 6. Нас интересует 90%-ный доверительный интервал, т. е. $\alpha = 0,1$. Таким образом, коэффициент использования памяти Nowon с вероятностью 0,9 попадает в интервал $[0,989; 0,991]$.

Задание 7.1. Для решения задачи о литейном цехе (пример 6.3) рассмотрите вариант модели с большим приоритетом для второго этапа и числом агрегатов 3. Определите длину переходного периода. Выполните 5 реплик эксперимента и для каждого из них получите коэффициент использования крана и коэффициент нагрузки операторов. Обработайте результаты в Excel. В тетради для лабораторных работ приведите таблицу Excel и снимок графика показателя, по которому был определен переходный период.

7.3. Сохраняемые величины и матрицы

Транзакты не могут непосредственно ссылаться друг на друга. Их «общение» реализуется через сохраняемые величины. Они могут рассматриваться как аналог переменных в языке программирования Паскаль. Одни транзакты могут записывать в них какие-то значения, а другие считывают или изменяют эти значения. Перед началом моделирования интерпретатор устанавливает значения сохраняемых величин в нуль. При необходимости можно задать ненулевые начальные значения сохраняемых величин с помощью команды

INITIAL A,B

Операнды имеют следующие значения:

A – имя сохраняемой величины или матрицы;

B – начальное значение, которое может быть UNCSPECIFIED, т. е. значения не определены. Это бывает необходимо при инициализации матрицы, которая будет содержать результаты экспериментов. Если матрица инициализирована неопределенными значениями, это позволяет процедуре ANOVA определить пропущенные значения.

Стандартный числовой атрибут X_j ($X\$<\text{имя}>$) позволяет обратиться к значению сохраняемой величины, например:

INITIAL X\$ZAPAS,40

Сохраняемая величина ZAPAS получает начальное значение 40.

Сохраняемая величина изменяется при входе транзакта в блок

SAVEVALUE A,B

Операнды имеют следующие значения:

A – номер (имя) сохраняемой величины, за которым может идти знак «+» или «-»;

B – присваиваемое значение: если нет знака после аргумента A, то оно замещает предыдущее значение сохраняемой величины A; при наличии знака предыдущее значение увеличивается (уменьшается) на аргумент B.

Примеры

SAVEVALUE ZAPAS+,20

Значение сохраняемой величины ZAPAS увеличивается на 20.

SAVEVALUE ZAPAS,60

Значение сохраняемой величины ZAPAS становится равной 60.

Аналогично можно объявить матричную сохраняемую величину (матрицу). Если для объявления одномерной сохраняемой величины не нужны специальные команды, интерпретатор определяет ее наличие по контексту, то для объявления матрицы используется команда

NAME MATRIX A,B,C,D,E,F,G

Значения операндов следующие:

NAME – имя матрицы;

A – не используется;

B – количество элементов матрицы по первому измерению;

C–G – количество элементов по второму–шестому измерениям.

Пример

Invent MATRIX ,1000,5

Объявляется матрица с именем Invent, состоящая из 1 000 строк и 5 столбцов.

Для обращения к элементам матрицы используется стандартный числовой атрибут `MX$<имя>(i,j,...)`. Например, `MX$Invent(3,4)` – значение элемента матрицы `Invent` в третьей строке, четвертом столбце.

Дать начальные значения матрице можно также командой `INITIAL`:

```
INITIAL MX$Invent(200,2),10
```

Элементу матрицы `Invent` в строке 200, столбце 2 присваивается значение 10.

```
INITIAL Invent,UNSPECIFIED
```

Матрица `Invent` инициализируется «неопределенными» значениями.

Значение одного элемента матрицы изменяется при прохождении транзакта через блок

```
MSAVEVALUE A,B,C,D
```

Значения операндов следующие:

A – имя (номер) матрицы, за которым может идти «+» или «-»;

B – номер строки;

C – номер столбца;

D – присваиваемая величина.

Пример

```
MSAVEVALUE Invent+,3,2,6.55
```

Элемент третьей строки второго столбца матрицы `Invent` увеличивается на 6.55.

Команда `RESET` не влияет на сохраняемые величины, скалярные и матричные.

Команда `CLEAR` может использоваться в двух режимах:

- `CLEAR ON` (по умолчанию) – очищает статистику, удаляет транзакты из цепей, обнуляет сохраняемые величины, матрицы и логические ключи;

- `CLEAR OFF` – то же самое, но не изменяет сохраняемые величины, матрицы и логические ключи.

Поэтому, если матрица используется для сбора статистики по различным прогонам, нужно использовать команду `CLEAR OFF`, чтобы не затирать статистику при каждом прогоне.

Для просмотра матриц используется диалоговое окно, которое вызывается командой **Window/Simulation Window/Matrix Window**.

7.4. Командные файлы

В GPSS можно использовать командные файлы, которые хранятся в текстовом формате (с расширением `*.txt`) и состоят из команд GPSS, организующих процесс постановки эксперимента. Для использования командного файла его нужно подключить с помощью команды **INCLUDE “имя файла”** в диалоге *Command/Custom*.

Пример 7.3. Организация экспериментов на модели трикотажной фабрики с помощью командного файла. Рассматривается моделирование работы трикотажной фабрики (пример 4.5) с тремя и четырьмя рабочими. Для каждого случая будет проведено по 7 реплик с различными установками генератора случайных чисел `RN1`. При этом каждая реплика включает переходный период в 1,5 тыс. ед. времени, статистика которого отбрасывается, и период стационарной работы в 3 тыс. ед. времени, по которому собирается статистика использования устройства `Nowon`. При этом вывод отчета будет отменен, а коэффициент использования устройства будет получен как значение `C4A SR$Nowon`. Результаты моделирования для каждой реплики будут сводиться в глобальную матрицу `RESULTS` с размерностью 2 строки на 7 столбцов. Выполняется однофакторный эксперимент по количеству рабочих с двумя уровнями фактора (3 и 4) и семью репликами на каждом уровне. Количество машин в данном примере не изменяется.

Создадим текстовый файл и сохраним его с именем *MyPlan.txt* (рис. 99). При создании этого файла можно использовать процедуру копирования.

```
RESULTS MATRIX      ,2,7      ;Матрица результатов
INITIAL RESULTS,UNSPECIFIED ;Инициализация матрицы
RMULT      1          ;Установка нач. значения RN1
START      1,NP       ;Моделирование переход. периода
RESET      ;Сброс стат. переход. периода
START      2,NP       ;Моделирование стац. режима
MSAVEVALUE RESULTS,1,1,SR$Nowon ;Зап. в matr. рез-в
CLEAR      OFF        ;Очистка с сохр. матрицы рез-в
RMULT      2
START      1,NP
RESET
START      2,NP
MSAVEVALUE RESULTS,1,2,SR$Nowon
... ..
```

```

MSAVEVALUE RESULTS,1,7,SR$Nowon
CLEAR OFF
Men STORAGE 4 ;Изменение количества рабочих
RMULT 1
START 1,NP
RESET
START 2,NP
MSAVEVALUE RESULTS,2,1,SR$Nowon
CLEAR OFF
... ..

```

Рис. 99. Командный файл *MyPlan.txt*

Затем откроем файл с программой моделирования трикотажной фабрики *Пример45.gps* (этот файл должен быть отлажен заранее) и откомпилируем модель – откроется журнал. Затем введем команду **Command/Custom** и в диалоговом окне напечатаем

INCLUDE “MyPlan.txt”

Моделирование начнется автоматически, будет выполнено 14 реплик экспериментов.

По окончании моделирования заполнится матрица RESULTS. Просмотреть ее можно командой **Window/Simulation Window/Matrix Window**.

Задание 7.2. Создайте файл *MyPlan.txt* и примените его для проведения экспериментов, как описано в примере 7.2. Просмотрите матрицу результатов.

7.5. Процедура ANOVA

Эта процедура предназначена для анализа результатов экспериментов. Она позволяет провести факторный анализ, причем можно использовать до шести факторов и до трех степеней взаимодействий между ними. Например, если в задаче о трикотажной фабрике изменить количество рабочих, то получим только один фактор (A), а если изменить еще и количество машин, то получим два фактора (A и B). Между ними может быть одно взаимодействие второй степени (AB). Если добавить третий фактор (C), например, варьировать среднее время до отказа, то возможна вторая степень взаимодействия (AB, AC, BC) или третья (ABC). Процедура ANOVA позволяет оценить степень значимости каждого фактора, или взаимодействия, а также определить для каждого из них среднее значение по всем репликам и 95%-ный доверительный интервал.

Процедура имеет три параметра, например:

ANOVA(A,B,C)

Параметры имеют следующие значения:

A – имя матрицы результатов;

B – размерность (количество измерений) матрицы результатов;

C – степень взаимодействия факторов, включаемая в отчет. Если этот аргумент равен 0 или 1, то взаимодействия в отчет не включаются, если 2 – то включаются взаимодействия второй степени, а если 3 и более – взаимодействия третьей степени.

Вызвать процедуру ANOVA можно через команду **Command/SHOW** и затем в окне ввода команд набрать

SHOW ANOVA(RESULTS,2,0)

Третий параметр в данном случае может быть любой, поскольку при одном факторе нет взаимодействий.

В окне *Журнал* выводится итоговая таблица работы процедуры ANOVA. Во второй части этой таблицы для каждого значения фактора A (количества рабочих) указывается следующая информация:

- Count – количество реплик;
- Mean – среднее значение (оценка матожидания);
- Minimum и Maximum – минимальное и максимальное наблюдавшиеся значения по всем репликам;
- 95% C.I. (SE) – 95%-ный доверительный интервал, в который попадает математическое ожидание.

Первая часть таблицы позволяет оценить степень влияния исследуемого фактора на значение показателя, представленного в таблице. В столбце F выводится значение критерия Фишера, рассчитанное по результатам эксперимента. В столбце *Critical Value of F (p=0.05)* приводится теоретическое значение этого критерия. Если расчетное значение больше F критического, то фактор считается *статистически значимым*.

Задание 7.3. Используйте процедуру ANOVA() для обработки результатов по модели трикотажной фабрики, которые ранее были сведены в таблицу RESULTS. Сравните, совпадают ли полученные ANOVA() оценки с теми, которые Вы самостоятельно рассчитали в Excel. Укажите, является ли фактор количества рабочих значимым в данном примере. Отчет ANOVA() и свои выводы приведите в тетради для лабораторных работ.

7.6. Процедура Experiment()

Язык PLUS включает в себя не только набор стандартных процедур (например, процедуры-функции формирования случайных чисел с экспоненциальным или нормальным распределением), но и позволяет определять процедуры пользователя. Особый класс процедур пользователя называется EXPERIMENT (*Эксперимент*). Эксперимент позволяет записать на языке PLUS программу управления прогонами и обработки результатов экспериментов. Программа *Эксперимент* может быть составлена непосредственно программистом либо автоматически – с помощью диалога через меню *Edit/Insert Experiment*. Рассмотрим первый способ проектирования программы *Эксперимент*.

Управление прогонами реализуется с помощью встроенной процедуры DoCommand(<строка>). Аргумент-строка транслируется в глобальном контексте и выполняется как команда. Если внутри аргумента-строки нужно выделить строку, ее берут в двойные кавычки.

Для выполнения эксперимента нужно задать команду CONDUCT с одним аргументом, именем эксперимента, и параметрами в круглых скобках.

Пример 7.4. Программа эксперимента для модели трикотажной фабрики. При исследовании модели из примера 4.5 будем изменять количество рабочих от значения параметра LeastMen до значения MostMen с шагом 1 (рис. 100). Для каждого значения этого фактора выполняется 6 прогонов с различными потоками случайных чисел (реплик). Результаты заносятся в глобальную матрицу Result с размерностью 3 × 6, т. е. эта матрица предусмотрена для трех значений фактора. Например, количество рабочих равно двум, трем и четырем, либо трем, четырем и пяти, либо семи, восьми и девяти и т. д. Переменная Level означает уровень фактора и всегда будет равна одному, двум или трем. Переменная KolMen означает текущее количество рабочих, для которого выполняются реплики. Переменная Replic означает номер реплики. После проведения прогонов и заполнения таблицы Result вызывается процедура ANOVA.

```
Result MATRIX          , 3, 6
      INITIAL          Result, UNSPECIFIED
EXPERIMENT BestMen (LeastMen, MostMen)
BEGIN
  Level=1;
  KolMen=LeastMen;
  WHILE ( (KolMen<=MostMen) 'AND' (Level<=3) ) DO
    BEGIN
      Replic=1;
      WHILE (Replic<=6) DO
        BEGIN
          RandBeg=11#Replic;
          DoCommand("CLEAR OFF");
          DoCommand(Catenate("RMULT ", RandBeg));
          DoCommand(Catenate("Men STORAGE ", KolMen));
          DoCommand("START 1, NP");
          DoCommand("RESET");
          DoCommand("START 2, NP");
          Result[Level, Replic]=SR$Nowon;
          Replic=Replic+1;
        END;
        Level=Level+1;
        KolMen=KolMen+1;
      END;
      ANOVA(Result, 2, 1);
    END;
```

Рис. 100. Программа автоматизации эксперимента для модели трикотажной фабрики

Процедура эксперимента вставляется в файл модели перед самой программой модели на языке GPSS. Затем выполняется компиляция модели, по окончании которой интерпретатор сообщает о регистрации процедуры BestMen. Если запустить программу просто командой START, то процедура эксперимента никакого влияния не окажет. Начать эксперимент, т. е. запустить процедуру BestMen, можно командой **Command/CONDUCT**, а затем в командной строке указать CONDUCT BestMen(3,5). После этого автоматически выполнятся прогоны и в журнале будет выведен отчет ANOVA.

Задание 7.4. На модели литейного цеха (примера 6.3) выполните корректно однофакторный имитационный эксперимент по определению числа агрегатов, которые одновременно должен обслуживать подъемный кран (рассмотрите варианты от 3 до 7 агрегатов). Используйте вариант модели с большим приоритетом для второго этапа.

Для решения этой задачи напишите программу автоматизации эксперимента на языке PLUS. Для каждого значения фактора (числа агрегатов) в этой программе нужно предусмотреть по 10 реплик эксперимента с разными начальными значениями генератора случайных чисел. Каждая реплика должна включать

моделирование по достижению стационарного периода, сброс статистики и продолжение моделирования до окончания 40-часовой рабочей недели.

По окончании экспериментов результаты должны быть обработаны с помощью процедуры ANOVA().

Контрольные вопросы

1. Что означает элемент u_{ij} в матрице реализаций?
2. В чем отличия переходного и стационарного периодов процесса? Когда обычно ставится задача исследования стационарного периода? Всегда ли он существует?
3. Какие существуют подходы к ликвидации влияния на статистику результатов переходного периода?
4. Как определяется продолжительность переходного периода в GPSS?
5. Какова последовательность действий для реализации одной реплики эксперимента на модели, если требуется не учитывать данные переходного периода?
6. Приведите формулы расчета оценок математического ожидания и дисперсии по результатам экспериментов.
7. Приведите формулу расчета точности доверительного интервала. Что такое уровень значимости?
8. Для чего используется сохраняемая величина в GPSS? Как к ней обратиться? Как изменить ее значение?
9. Каково назначение команды INITIAL?
10. Как объявить и использовать матрицу в GPSS?
11. В чем отличие режимов команды CLEAR?
12. Для чего предназначены командные файлы? Как их использовать?
13. Понятия фактора и реплики эксперимента.
14. Для чего предназначена процедура ANOVA()? Что означают поля в ее отчете?
15. Для чего используется критерий Фишера в отчете процедуры ANOVA()?
16. Назначение и использование процедуры Experiment().

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. **Шрайбер, Т. Дж.** Моделирование на GPSS / Т. Дж. Шрайбер. – М. : Машиностроение, 1980.
2. **Кельтон, В.** Имитационное моделирование. Классика CS / В. Кельтон, А. Лоу. – 3-е изд. – СПб. : BHV, 2004.
3. **Кудрявцев, Е. М.** GPSS World. Основы имитационного моделирования различных систем / Е. М. Кудрявцев. – М. : ДМК Пресс, 2004.
4. **Рыжиков, Ю. И.** Имитационное моделирование. Теория и технологии / Ю. И. Рыжиков. – СПб. : КОРОНА принт, 2004.
5. **Основы** имитационного и статистического моделирования : учеб. пособие / Ю. С. Харин [и др.]. – Минск : Дизайн ПРО, 1997.
6. **Максимей, И. В.** Имитационное моделирование на ЭВМ / И. В. Максимей. – М. : Радио и связь, 1988.
7. **Емельянов, А. А.** Имитационное моделирование экономических процессов : учеб. пособие / А. А. Емельянов, Е. А. Власова, Р. В. Дума ; под ред. А. А. Емельянова. – 2-е изд., перераб. и доп. – М. : Финансы и статистика, 2006.
8. **Экономико-математические** методы и модели : практикум : в 5 ч. / авт.-сост. : В. В. Бондарева, Т. А. Заяц. – Гомель : Бел. торгово-экон. ун-т потребит. кооп., 2000. – Ч. 2.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
----------------	---

ЛАБОРАТОРНАЯ РАБОТА 1. ОЦЕНКА ИНВЕСТИЦИОННЫХ ПРОЕКТОВ МЕТОДОМ МОНТЕ-КАРЛО.....	5
1.1. Методы оценки инвестиционных проектов	5
1.2. Описание моделируемого объекта	7
1.3. Расчет чистой текущей стоимости проекта на основании наиболее вероятных значений	8
1.4. Применение метода Монте-Карло для оценки характеристик чистой текущей стоимости проекта	11
1.5. Задание для самостоятельной работы	16
Контрольные вопросы	16
ЛАБОРАТОРНАЯ РАБОТА 2. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКА МОДЕЛИРОВАНИЯ GPSS	17
2.1. Описание моделируемого объекта	17
2.2. События и таймер модельного времени.....	19
2.3. Объекты языка GPSS	22
2.4. Простейшие операторы GPSS.....	25
2.4.1. Введение транзактов в модель	25
2.4.2. Удаление транзактов из модели	27
2.4.3. Управление устройствами.....	28
2.4.4. Задержка транзактов во времени	29
2.4.5. Сбор статистики при ожидании в очереди	30
2.5. Модель работы парикмахерской.....	32
2.6. Логика работы моделирующей системы.....	35
2.7. Расчет статистических характеристик объектов	38
2.8. Задания для самостоятельной работы	40
Контрольные вопросы	44
ЛАБОРАТОРНАЯ РАБОТА 3. ТЕХНОЛОГИЯ РАБОТЫ В СИСТЕМЕ GPSS WORLD	46
3.1. Общие сведения о системе GPSS World.....	46
3.2. Работа с файлами	46
3.3. Ввод текста модели	47
3.4. Компиляция и исправление ошибок.....	49
3.5. Выполнение программы имитации	51
3.6. Структура стандартного отчета	53
3.7. Окно Блоков	57
3.8. Управляющие операторы GPSS World.....	62
3.9. Стандартные числовые атрибуты	64
3.10. Просмотр значений СЧА и выражений	65
3.11. Просмотр цепей текущих и будущих событий.....	67
3.12. Окно Графиков.....	70
3.13. Окна Устройств и Очередей.....	72
3.14. Задания для самостоятельной работы	74
Контрольные вопросы	76
ЛАБОРАТОРНАЯ РАБОТА 4. ПРИЕМЫ ПРОГРАММИРОВАНИЯ МОДЕЛЕЙ (НАЧАЛО).....	77
4.1. Организация обработки различных типов транзактов	77
4.2. Использование приоритетов транзактов	83
4.3. Изменение маршрутов движения транзактов	87
4.4. Моделирование многоканальных устройств	94
4.5. Примеры использования блока TRANSFER.....	101
4.6. Задания для самостоятельной работы	105
Контрольные вопросы	110
ЛАБОРАТОРНАЯ РАБОТА 5. ИСПОЛЬЗОВАНИЕ РАСПРЕДЕЛЕНИЙ ВЕРОЯТНОСТЕЙ В GPSS	111
5.1. Основные сведения о методах генерации случайных величин в GPSS World.....	111
5.2. Моделирование пуассоновских потоков событий.....	112
5.3. Моделирование эмпирических дискретных функций распределения.....	115
5.4. Моделирование эмпирических непрерывных функций распределения.....	118
5.5. Определение непрерывных равномерных функций.....	120
5.6. Задания для самостоятельной работы	121
Контрольные вопросы	123
ЛАБОРАТОРНАЯ РАБОТА 6. ПРИЕМЫ ПРОГРАММИРОВАНИЯ МОДЕЛЕЙ (ПРОДОЛЖЕНИЕ).....	124
6.1. Использование стандартных числовых атрибутов.....	124
6.2. Параметры транзактов и арифметические переменные.....	128
6.3. Изменение приоритета транзакта	132
6.4. Выбор элементов по их состояниям	137
6.5. Резидентное и транзитное время транзактов	144
6.6. Использование таблиц в языке GPSS	145

6.7. Задания для самостоятельной работы	150
Контрольные вопросы	155
ЛАБОРАТОРНАЯ РАБОТА 7. ОРГАНИЗАЦИЯ ИМИТАЦИОННЫХ	
ЭКСПЕРИМЕНТОВ.....	156
7.1. Переходный период стохастического процесса	156
7.2. Статистическая обработка результатов моделирования.....	161
7.3. Сохраняемые величины и матрицы	164
7.4. Командные файлы.....	166
7.5. Процедура ANOVA.....	168
7.6. Процедура Experiment().....	169
Контрольные вопросы	171
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ.....	173

Учебное издание

ОСНОВЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Практикум

**к лабораторным занятиям по курсам
«Имитационное моделирование информационных процессов
и систем» и «Имитационное моделирование экономических
процессов» для студентов дневной формы обучения
специальностей 1-26 03 01 «Управление информационными
ресурсами» и 1-25 01 07 «Экономика и управление
на предприятии» специализации 1-25 01 07 02
«Экономическая информатика»**

Автор-составитель
Еськова Оксана Ивановна

Редактор М. П. Любошенко
Технический редактор Н. Н. Короедова
Компьютерная верстка Л. Г. Макарова

Подписано в печать 15.09.09. Бумага типографская № 1.
Формат 60 × 84^{1/16}. Гарнитура Таймс. Ризография.
Усл. печ. л. 10,23. Уч.-изд. л. 10,40. Тираж 100 экз.
Заказ №

Учреждение образования
«Белорусский торгово-экономический
университет потребительской кооперации».
246029, г. Гомель, просп. Октября, 50.
ЛИ № 02330/0494302 от 04.03.2009 г.

Отпечатано в учреждении образования
«Белорусский торгово-экономический университет
потребительской кооперации».
246029, г. Гомель, просп. Октября, 50.